

# Linux VPN Client 3.4

## Administrator's Guide

TheGreenBow is a registered trademark.

Microsoft, Windows 10, and Windows 11 are either registered trademarks or brand names owned by Microsoft Corp. in the U.S.A. and/or in other countries.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Ubuntu and the Ubuntu logo are trademarks or registered trademarks of Canonical Group Ltd. in the United Kingdom, other countries, or both.

Red Hat, Red Hat Enterprise Linux, the Red Hat logo, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Debian is a registered trademark owned by Software in the Public Interest, Inc. in the United States, managed by the Debian project.

Apple, iPhone, iOS and macOS are trademarks of Apple Inc., registered in the U.S. and other countries and regions.

Android, Chrome, Google and Google Apps are trademarks or registered trademarks of Google LLC.

Any other third-party trademarks mentioned in this document are the property of their respective owners.

All reproduction rights are reserved, including for iconographic or photographic representations. No part of this document may be copied and/or published by any means whatsoever without our express written consent.

All the information provided is subject to modifications at any time and without notice.

Despite our utmost care and attention in producing this document and its regular updates, errors may have been introduced in the information provided. If you notice any issues, please feel free to contact us and let us know. We will make the necessary changes.

# Table of contents

<b>1</b>	<b>Overview .....</b>	<b>1</b>
1.1	Introduction .....	1
1.2	Security .....	1
1.3	Ergonomic.....	1
1.4	Simple.....	1
1.5	Universal.....	2
1.6	Features .....	2
1.7	What's new in release 3.4 .....	3
1.7.1	Features .....	3
1.7.2	Certificate authentication and revocation.....	3
<b>2</b>	<b>Downloading and verifying the software.....</b>	<b>4</b>
2.1	Introduction .....	4
2.2	Verification procedure in Red Hat.....	4
2.3	Verification procedure in Ubuntu.....	5
2.4	Technical information .....	7
2.4.1	Deleting the key in Red Hat.....	7
2.4.2	Deleting the key in Ubuntu .....	7
<b>3</b>	<b>Installing the software.....</b>	<b>8</b>
3.1	Introduction .....	8
3.2	Minimum requirements .....	8
3.3	Dependencies .....	8
3.4	Package contents.....	9
3.5	Installation procedure .....	10
3.5.1	In Red Hat .....	10
3.5.2	In Ubuntu.....	13
<b>4</b>	<b>Activating the software.....</b>	<b>15</b>
4.1	Introduction .....	15
4.2	Trial period .....	15
4.3	Format and content of the vpnsetup.json file.....	15

4.4	Activation procedure.....	16
4.4.1	Automatic activation.....	16
4.4.2	Manual activation.....	17
4.4.3	Activation successful.....	20
4.5	Activation errors .....	20
<b>5</b>	<b>tgbtray icon in the system menu.....</b>	<b>21</b>
5.1	Adding the icon to the system menu .....	21
5.1.1	In Red Hat.....	21
5.1.2	In Ubuntu.....	23
5.2	State of the system menu icon.....	23
5.3	Contextual menu of the icon in the system menu .....	24
<b>6</b>	<b>Updating the software.....</b>	<b>25</b>
6.1	Backing up before updating.....	25
6.2	Updating from the command line.....	25
<b>7</b>	<b>Uninstalling the software.....</b>	<b>27</b>
7.1	In Red Hat.....	27
7.2	In Ubuntu.....	27
<b>8</b>	<b>Using the test tunnel.....</b>	<b>29</b>
<b>9</b>	<b>Command line.....</b>	<b>30</b>
9.1	Introduction .....	30
9.2	Displaying help .....	30
9.3	Displaying the version of the software .....	31
9.4	Displaying the number of days remaining before the trial period expires .....	31
9.5	Listing the configured VPN connections.....	31
9.6	Opening a VPN connection .....	32
9.7	Closing a VPN connection.....	32
9.8	Displaying the status of the VPN connection.....	33
9.9	Setting a PIN code .....	33
9.10	Resetting the tgbray icon.....	33
9.11	Resetting the IKE daemon.....	34

<b>10</b>	<b>Configuring VPN connections.....</b>	<b>35</b>
10.1	Introduction .....	35
10.2	Using IPsec DR mode.....	35
10.3	Protecting the VPN configuration.....	36
10.4	Managing certificates.....	36
10.5	Updating the VPN configuration .....	37
<b>11</b>	<b>Using tokens and smart cards .....</b>	<b>38</b>
11.1	Introduction .....	38
11.2	vpnconf.ini file .....	38
11.3	Installing the middleware .....	40
11.4	Creating the vpnconf.ini file .....	41
<b>12</b>	<b>Logs.....</b>	<b>42</b>
12.1	Introduction .....	42
12.2	Exporting in text format .....	42
12.3	Defining the log level .....	43
12.3.1	Displaying the current log level.....	43
12.3.2	Changing the log level.....	44
<b>13</b>	<b>Configuring NetworkManager .....</b>	<b>45</b>
<b>14</b>	<b>Automatic startup .....</b>	<b>46</b>
14.1	Starting tgbray automatically .....	46
14.1.1	In Red Hat .....	46
14.1.2	In Ubuntu.....	48
14.2	Opening the tunnel automatically .....	50
<b>15</b>	<b>Selecting the kernel.....</b>	<b>51</b>
<b>16</b>	<b>Current limitations .....</b>	<b>52</b>
<b>17</b>	<b>Managing errors.....</b>	<b>53</b>
17.1	User must belong to “tgb” group .....	53
17.2	Cannot get VPN connection list .....	53
17.3	Opening a VPN connection failed .....	53
17.4	Non-root users must not be able to access the configuration file.....	54

17.5	Checking the driver .....	54
17.6	Cannot start IKE daemon .....	55
17.7	IKE daemon is unresponsive.....	56
17.8	Token or smart card errors .....	56
17.9	Virtual machine does not recognize a token or smart card.....	57
<b>18</b>	<b>Related reference documents .....</b>	<b>58</b>
<b>19</b>	<b>OpenSSL license.....</b>	<b>59</b>
<b>20</b>	<b>Contact .....</b>	<b>62</b>
20.1	Information.....	62
20.2	Sales.....	62
20.3	Support .....	62

## Document revision history

Version	Date	Sections/pages concerned	Description of change	Author
1.0	2024-02-07	All	Initial release	EBO, FBO, BB
1.1	2024-09-23	3.3	Added details about dependencies	EBO, BB
1.2	2025-04-01	2.2 & 2.3	Updated the software verification procedures	EBO, BB
		3.3	Added a warning notice about dependencies in Red Hat	
		6	Restructured the updating procedure and added details about updating to version 3.4.4 or higher	
		13	Added chapter on how to configure NetworkManager to run scripts	
		14	Added section on starting tgbtray automatically in Red Hat	
			Added section on starting tunnel automatically	
		15	Added chapter on kernel selection	
1.3	2025-06-13	3.3	Added details about commands for <code>systemd-resolved</code> service	AVE, EBO, FBO, BB
		5.3	Added details regarding multiple connections	
		10.1	Corrected configuration file extension and added details regarding use of multiple configuration files	
		10.2	Added a section with details about working with the IPsec DR mode	
		12.3	Added a section about how to display and change the log level	
		17.6	Updated screenshot	

---

# 1 Overview

## 1.1 Introduction

Thank you for downloading our Linux VPN Client 3.4 software.

The Linux VPN Client has been thoughtfully designed to address the needs of major corporations, critical market operators, as well as civil and government bodies. It provides a high level of communication security and is also easy to deploy, integrate, and use.

Users of the Linux VPN Client also benefit from highly personal support that goes from customer-specific follow-up to the integration of customized developments.

It does not require the existing key management infrastructure (PKI) to be reconsidered and it is designed to be transparently integrated into the IKEv2 gateways that have been set up.

The Linux VPN Client is marketed on the basis of an annual subscription. The subscription includes customer-specific support and software maintenance.

## 1.2 Security

Specifically designed for nomadic work practices, the Linux VPN Client is an IKEv2 IPsec VPN client software for Linux workstations that enables users to establish perfectly secure connections to the company's information system over the internet. It implements a broad range of encryption and hashing algorithms, as well as various strong authentication methods.

## 1.3 Ergonomic

Easy to install, easy to configure and deploy, perfectly transparent to the user, the Linux VPN Client is widely recognized today for its unparalleled ergonomics.

## 1.4 Simple

Our configuration guides make integration and deployment tasks painless by speeding up the implementation of an end-to-end VPN solution.





## 1.5 Universal

The Linux VPN Client runs on Ubuntu 22.04 (kernel 5.x) & 24.04 (kernel 6.x) and Red Hat 9.4/9.5/9.6. The software is compatible with a great number of IPsec firewalls/gateways available on the market. The constantly growing list of firewalls/gateways that have been tested in our laboratory is available on [TheGreenBow](https://thegreenbow.com)'s website.

## 1.6 Features

- IPsec network driver and IKE module developed by TheGreenBow
- IPsec stack integrated in Linux kernel
- Support for the IKEv2 protocol
- Interoperable with all IKEv2 compatible VPN firewalls/gateways
- Encryption: 128 / 192 / 256-bit AES CBC, CTR and GCM
- Hashing: SHA-2 256/384/512
- DH groups: 14-21, 28
- X.509 certificate management: PEM/PFX, PKCS #12<sup>1</sup>
- Authentication: preshared key, certificates, EAP, two-factor authentication (certificate + EAP)
- IP fragmentation
- "All traffic through the VPN tunnel" mode
- Dead Peer Detection (DPD): Detection of gateway traffic interruption
- Redundant gateway
- CP mode (Configuration Payload)
- Automatic negotiation of algorithms with gateway
- IKEv2 fragmentation
- Automatic NAT-Traversal mode
- Local ID, Remote ID
- Import VPN configurations generated using TheGreenBow Windows Enterprise VPN Client
- Control from the command line or using the graphical interface
- Activation using a license
- Support for syslog event log format and protocol
- Two packages, each one compatible with one of the following Linux distributions:
  - Red Hat version 9.4/9.5/9.6, 64-bit
  - Ubuntu 22.04 (kernel 5.x) & 24.04 (kernel 6.x), 64-bit
- Integrated in system menu (systray)

---

<sup>1</sup> Configuration to be performed using the Windows Enterprise VPN Client.

## 1.7 What's new in release 3.4

### 1.7.1 Features

- Compatible with most IPsec and SSL gateways, including those that support the IPsec DR (Restricted) repository
- Support for RFC 6023 (Childless IKE Initiation) for enhanced security
- Compatible with Red Hat 9.4/9.5/9.6 and Ubuntu 22.04/24.04
- Create scripts using Gnome NetworkManager

### 1.7.2 Certificate authentication and revocation

Due to increased security requirements, deprecation of certain algorithms, and stricter rules for using certificates, version 3.4 of the Linux VPN Client comes with certain restrictions on certificates.

- Support for the following certificate authentication methods:
  - Method 1: RSA Digital Signature with SHA-2 [RFC 7296]
  - Method 9: ECDSA on the secp256r1 curve with SHA-2 (256 bits) [RFC 4754]
  - Method 10: ECDSA on the secp384r1 curve with SHA-2 (384 bits) [RFC 4754]
  - Method 11: ECDSA on the secp521r1 curve with SHA-2 (512 bits) [RFC 4754]
  - Method 14: Digital Signature Authentication RSASSA PSS with SHA-2 (256/384/512 bits) [RFC 7427]
  - Method 214: ECDSA "BrainpoolP256r1" with SHA-2 (256 bits) (only available for gateways that support this method)
- Certificate authentication method 14, which is based on the RSASSA-PSS signature algorithm, is used by default for all RSA certificates
- End of support for Method 1: RSA Digital Signature with SHA-1 [RFC 7296]
- RSA certificates with less than 2048-bit key length are rejected
- ECDSA certificates with less than 256-bit key length are rejected
- Key Usage and Extended Key Usage of certificates is verified



## 2 Downloading and verifying the software

### 2.1 Introduction

The Linux VPN Client is available for download on [TheGreenBow](https://thegreenbow.com)'s website.

Prior to installing the Linux VPN Client, it is essential to verify the authenticity of the software package downloaded from our website in order to confirm that it has indeed been signed by TheGreenBow and that it has not been altered in any way.

### 2.2 Verification procedure in Red Hat

To verify the authenticity of the Red Hat package, follow the steps below:

1. Open a terminal window.
2. Run the following command to download the public key:

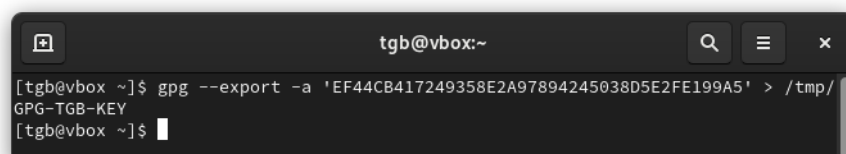
```
gpg --keyserver keys.openpgp.org --recv-keys  
EF44CB417249358E2A97894245038D5E2FE199A5
```



```
tgb@vbox:~  
[tgb@vbox ~]$ gpg --keyserver keys.openpgp.org --recv-keys EF44CB417249358E2A97894245038D5E2FE199A5  
gpg: directory '/home/tgb/.gnupg' created  
gpg: keybox '/home/tgb/.gnupg/pubring.kbx' created  
gpg: /home/tgb/.gnupg/trustdb.gpg: trustdb created  
gpg: key 45038D5E2FE199A5: public key "TGB Linux product manager <linux@thegreenbow.com>" imported  
gpg: Total number processed: 1  
gpg: imported: 1  
[tgb@vbox ~]$
```

3. Run the following command to export the key to a temporary file:

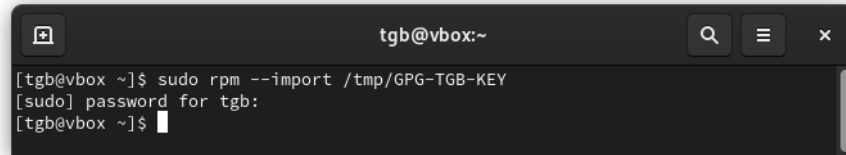
```
gpg --export -a  
'EF44CB417249358E2A97894245038D5E2FE199A5' > /tmp/GPG-  
TGB-KEY
```



```
tgb@vbox:~  
[tgb@vbox ~]$ gpg --export -a 'EF44CB417249358E2A97894245038D5E2FE199A5' > /tmp/GPG-TGB-KEY  
[tgb@vbox ~]$
```

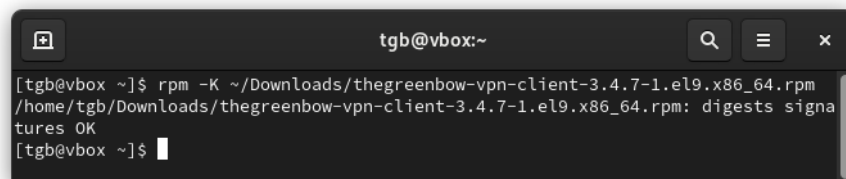
4. Run the following command to import the key:

```
sudo rpm --import /tmp/GPG-TGB-KEY
```



5. Verify the software package by running the following command in the directory where the package is located (Downloads in this case, where **r** stands for the software revision number and **b** the software build number):

```
rpm -K ~/Downloads/thegreenbow-vpn-client-3.4.r-  
b.el9.x86_64.rpm
```



6. Make sure that the output data is as follows:

```
/home/[username]/Downloads/thegreenbow-vpn-client-3.4.r-  
b.el9.x86_64.rpm: digests signatures OK
```

If this is not the case, contact customer support:

<https://www.thegreenbow.com/en/support/online-support/technical-support/>.

## 2.3 Verification procedure in Ubuntu

To verify the authenticity of the Ubuntu package, follow the steps below:

1. Open a terminal window (Ctrl + Alt + T).
2. Run the following command to download the public key and import it into the local GPG key store:

```
gpg --keyserver keys.openpgp.org --recv-keys  
EF44CB417249358E2A97894245038D5E2FE199A5
```



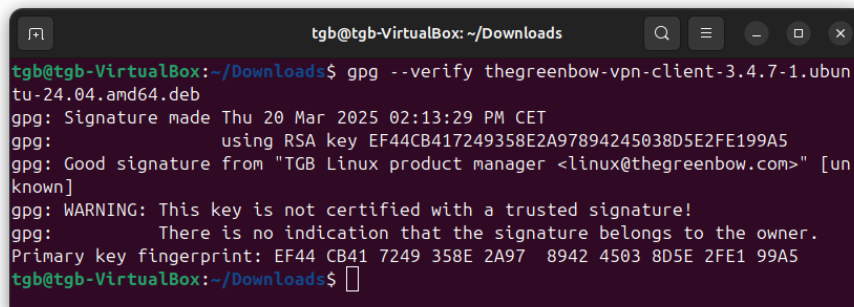
```
tgb@TGB-Ubuntu-VM: ~/Downloads  
tgb@TGB-Ubuntu-VM:~/Downloads$ gpg --keyserver keys.openpgp.org --recv-keys EF44  
CB417249358E2A97894245038D5E2FE199A5  
gpg: directory '/home/tgb/.gnupg' created  
gpg: keybox '/home/tgb/.gnupg/pubring.kbx' created  
gpg: /home/tgb/.gnupg/trustdb.gpg: trustdb created  
gpg: key 45038D5E2FE199A5: public key "TGB Linux product manager <linux@thegreen  
bow.com>" imported  
gpg: Total number processed: 1  
gpg: imported: 1  
tgb@TGB-Ubuntu-VM:~/Downloads$
```

3. Verify the software package by running the following command in the directory where the package is located (replace the **r** with the software package revision number and the **b** with the software package build number):

```
gpg --verify thegreenbow-vpn-client-3.4.r-b.ubuntu-  
24.04.deb
```

4. Make sure that the output data is as follows:

```
Good signature from "TGB Linux product manager  
linux@thegreenbow.com"
```



```
tgb@tgb-VirtualBox: ~/Downloads  
tgb@tgb-VirtualBox:~/Downloads$ gpg --verify thegreenbow-vpn-client-3.4.7-1.ubun  
tu-24.04.amd64.deb  
gpg: Signature made Thu 20 Mar 2025 02:13:29 PM CET  
gpg: using RSA key EF44CB417249358E2A97894245038D5E2FE199A5  
gpg: Good signature from "TGB Linux product manager <linux@thegreenbow.com>" [un  
known]  
gpg: WARNING: This key is not certified with a trusted signature!  
gpg: There is no indication that the signature belongs to the owner.  
Primary key fingerprint: EF44 CB41 7249 358E 2A97 8942 4503 8D5E 2FE1 99A5  
tgb@tgb-VirtualBox:~/Downloads$
```

If this is not the case, contact customer support:

<https://www.thegreenbow.com/en/support/online-support/technical-support/>.

## 2.4 Technical information

The installation package is signed with a 4096-bit RSA key. The corresponding public key is available here:

<https://keys.openpgp.org/vks/v1/by-fingerprint/EF44CB417249358E2A97894245038D5E2FE199A5>.

Key ID: EF44 CB41 7249 358E 2A97 8942 4503 8D5E 2FE1 99A5.

Key fingerprint: 2FE199A5.

### 2.4.1 Deleting the key in Red Hat

1. Run the following command to retrieve the key's full identifier:

```
rpm -q gpg-pubkey --qf  
'%{NAME}-%{VERSION}-%{RELEASE}\t%{SUMMARY}\n'
```

The output data should specify the name of the key along with its version and release numbers as follows:

```
gpg-pubkey-2fe199a5-[revision_number]    TGB Linux  
product manager linux@thegreenbow.com public key
```

2. Run the following command to delete the key, making sure to replace the release number with the number retrieved in the previous step:

```
sudo rpm --erase gpg-pubkey-2fe199a5-[revision_number]
```

### 2.4.2 Deleting the key in Ubuntu

Run the following command to delete the public key from the local GPG key store:

```
gpg --delete-key  
EF44CB417249358E2A97894245038D5E2FE199A5
```



## 3 Installing the software

### 3.1 Introduction

After having downloaded the Linux VPN Client from TheGreenBow's website and having verified its authenticity (see chapter 2 Downloading and verifying the software), you can install the program from the command line.

### 3.2 Minimum requirements

To install the Linux VPN Client you must have superuser privileges (or root access) on the machine.

In addition, you will need to create a configuration file for the Linux workstation using the Windows Enterprise VPN Client.

### 3.3 Dependencies

When you install the Linux VPN Client, the installer checks whether the following dependencies are available:

- `dkms`<sup>1</sup> and `systemd-resolved`<sup>2</sup> in Red Hat
- `dkms`<sup>3</sup> in Ubuntu



On Red Hat systems, administrators must manually install the EPEL package, which contains the DKMS dependency, or add DKMS to an internal repository. To accommodate users who prefer not to rely on external repositories, the Linux VPN Client does not include EPEL as a dependency. For more information about EPEL, see [Red Hat's EPEL documentation](#).



If any of the above packages are missing during installation, the VPN Client will not be installed, and an error message will specify the missing packages.

<sup>1</sup> The DKMS dependency is offered in the EPEL repository. Administrators who do not want to add external dependencies to their network can work around installing EPEL, by adding DKMS in an internal repository.

<sup>2</sup> This package is required for DNS support on a virtual interface.

<sup>3</sup> This package is installed by default in Ubuntu.

To check whether `systemd-resolved` is available in Red Hat, execute either of the following commands:

```
sudo systemd-resolve --status
```

```
sudo systemctl status systemd-resolved.service
```

To install `systemd-resolved` in Red Hat, execute the following commands successively:

```
sudo dnf install systemd-resolved
sudo systemctl enable systemd-resolved.service
ln -sf /run/systemd/resolve/stub-resolv.conf
/etc/resolv.conf
```



If `systemd-resolved` is available but not enabled, simply execute the last two commands above.

## 3.4 Package contents

When you install the Linux VPN Client, the following directories and files will be added to the workstation:

- `/usr/bin/tgbtray`: program that manages the Linux VPN Client's icon in the system menu (systray)
- `/usr/bin/tgbctl`: command used to control the Linux VPN Client from the command line
- `/usr/sbin/tgbiked`: Linux VPN Client daemon running in the background
- `/lib/systemd/system/tgbiked.service`: daemon configuration file
- `/etc/tgb/conf.tgb`: VPN configuration file, including a TheGreenBow test tunnel
- `/etc/tgb/vpnsetup.json`: license file for the Linux VPN Client
- `/usr/share/doc/thegreenbow/CLUF_VPN_TheGreenBow_vFR3.51.pdf`: document containing TheGreenBow's End User License Agreement
- `/usr/share/icons/thegreenbow`: folder containing the icons used by `tgbtray`
- `/usr/share/applications/thegreenbow.desktop`: application launcher
- `/usr/src/tgbtun-1.2`: folder containing the source files for dynamic kernel module support (DKMS)



## 3.5 Installation procedure

The Linux VPN Client must be installed from the command line.



For Red Hat, refer to section 3.5.1 In Red Hat.



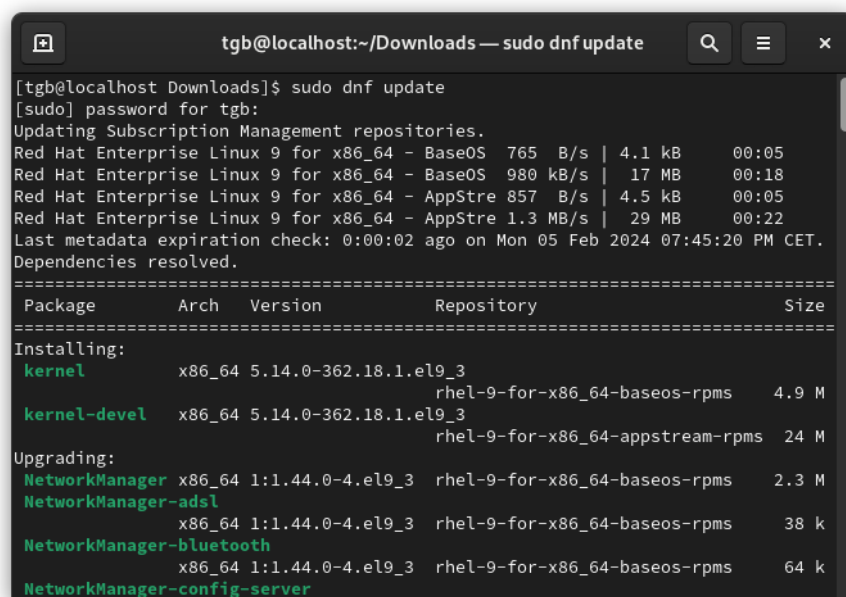
For Ubuntu, refer to section 3.5.2 In Ubuntu.

### 3.5.1 In Red Hat

To install the Linux VPN Client in Red Hat, proceed as follows:

1. If you downloaded the software package on a machine other than the one on which the Linux VPN Client is to be installed, copy it to the destination machine.
2. Open a terminal window.
3. Run the following command to update the package repositories:

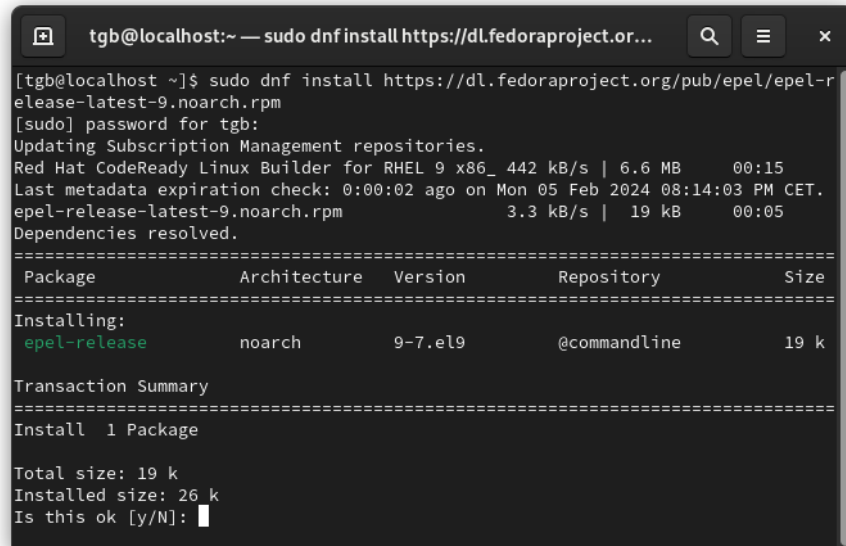
```
sudo dnf update
```



```
tgb@localhost:~/Downloads — sudo dnf update
[tgb@localhost Downloads]$ sudo dnf update
[sudo] password for tgb:
Updating Subscription Management repositories.
Red Hat Enterprise Linux 9 for x86_64 - BaseOS 765 B/s | 4.1 kB 00:05
Red Hat Enterprise Linux 9 for x86_64 - BaseOS 980 kB/s | 17 MB 00:18
Red Hat Enterprise Linux 9 for x86_64 - AppStre 857 B/s | 4.5 kB 00:05
Red Hat Enterprise Linux 9 for x86_64 - AppStre 1.3 MB/s | 29 MB 00:22
Last metadata expiration check: 0:00:02 ago on Mon 05 Feb 2024 07:45:20 PM CET.
Dependencies resolved.
=====
Package      Arch  Version                      Repository                      Size
=====
Installing:
kernel       x86_64 5.14.0-362.18.1.el9_3        rhel-9-for-x86_64-baseos-rpms  4.9 M
kernel-devel x86_64 5.14.0-362.18.1.el9_3        rhel-9-for-x86_64-appstream-rpms 24 M
Upgrading:
NetworkManager x86_64 1:1.44.0-4.el9_3            rhel-9-for-x86_64-baseos-rpms  2.3 M
NetworkManager-adsl x86_64 1:1.44.0-4.el9_3            rhel-9-for-x86_64-baseos-rpms   38 k
NetworkManager-bluetooth x86_64 1:1.44.0-4.el9_3            rhel-9-for-x86_64-baseos-rpms   64 k
NetworkManager-config-server
```

4. Run the following command to install the extra package for Enterprise Linux<sup>1</sup>:

```
sudo dnf install
https://dl.fedoraproject.org/pub/epel/epel-release-
latest-9.noarch.rpm
```

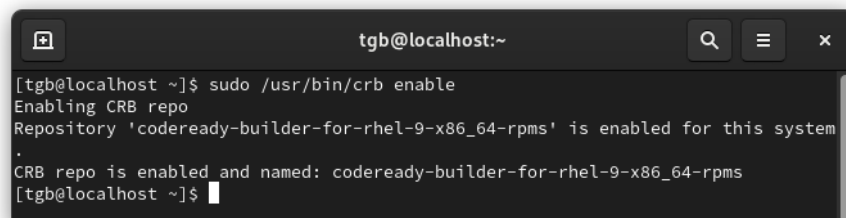


A terminal window titled 'tgb@localhost:~ — sudo dnf install https://dl.fedoraproject.or...' showing the command 'sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm' being executed. The output shows the package being installed from the @commandline repository. A table lists the package details: Package (epel-release), Architecture (noarch), Version (9-7.el9), Repository (@commandline), and Size (19 k). The transaction summary shows 1 package being installed, with a total size of 19 k and an installed size of 26 k. The prompt 'Is this ok [y/N]:' is visible at the bottom.

Package	Architecture	Version	Repository	Size
epel-release	noarch	9-7.el9	@commandline	19 k

5. Run the following command to enable the CRB repository:

```
sudo /usr/bin/crb enable
```



A terminal window titled 'tgb@localhost:~' showing the command 'sudo /usr/bin/crb enable' being executed. The output indicates that the CRB repository 'codeready-builder-for-rhel-9-x86\_64-rpms' is enabled for this system. The prompt '[tgb@localhost ~]\$' is visible at the bottom.

6. Run the following command again to install the dkms package:

```
sudo dnf install dkms
```

<sup>1</sup> If EPEL cannot be installed on the workstation, the dkms, libappindicator3, and libdbusmenu packages must be placed in the internal repository.

```
tgb@localhost:~ — sudo dnf install dkms

[tgb@localhost ~]$ sudo dnf install dkms
Updating Subscription Management repositories.
Last metadata expiration check: 0:00:49 ago on Mon 05 Feb 2024 08:16:24 PM CET.
Dependencies resolved.
=====
Package      Arch    Version      Repository      Size
=====
Installing:
  dkms        noarch  3.0.12-1.el9  epel             80 k
Installing dependencies:
  kernel-devel-matched
  x86_64      5.14.0-362.18.1.el9_3  rhel-9-for-x86_64-appstream-rpms 4.9 M
=====
Transaction Summary
=====
Install 2 Packages

Total download size: 5.0 M
Installed size: 173 k
Is this ok [y/N]:
```

7. Access the folder containing the thegreenbow-vpn-client-3.4.**r**-**b**.el9.x86\_64.rpm package (where **r** is the software revision number and **b** the software build number).
8. Run the following command to install the Linux VPN Client software (where **r** is the software revision number and **b** the software build number):

```
sudo dnf install thegreenbow-vpn-client-3.4.r-b.el9.x86_64.rpm
```

```
tgb@localhost:~/Downloads — sudo dnf install thegreenbow-v...

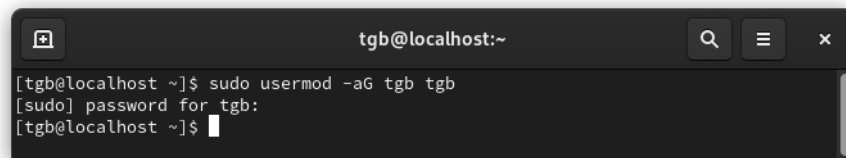
[tgb@localhost Downloads]$ sudo dnf install thegreenbow-vpn-client-3.4.2-1.el9.x86_64.rpm
[sudo] password for tgb:
Updating Subscription Management repositories.
Last metadata expiration check: 0:03:18 ago on Mon 05 Feb 2024 08:16:24 PM CET.
Dependencies resolved.
=====
Package      Arch    Version      Repository      Size
=====
Installing:
  thegreenbow-vpn-client
  x86_64      3.4.2-1.el9  @commandline    3.3 M
Installing dependencies:
  libappindicator-gtk3  x86_64  12.10.0-33.el9  epel            41 k
  libdbusmenu           x86_64  16.04.0-19.el9  epel            134 k
  libdbusmenu-gtk3      x86_64  16.04.0-19.el9  epel            40 k
  libindicator-gtk3     x86_64  12.10.1-22.el9  epel            66 k
  pcsc-lite-libs        x86_64  1.9.4-1.el9     rhel-9-for-x86_64-baseos-rpms 30 k
  systemd-resolved      x86_64  252-18.el9     rhel-9-for-x86_64-baseos-rpms 369 k
=====
Transaction Summary
=====
Install 7 Packages

Total size: 4.0 M
Total download size: 679 k
Installed size: 15 M
Is this ok [y/N]:
```

9. Add the VPN users to the `tgb` group by running the following command:

```
sudo usermod -aG tgb $(whoami)
```

10. Enter the administrator's password and press Enter.



11. If you use Safenet/Gemalto smart cards, run the following command:

```
sudo dnf install SafenetAuthenticationClient-10.8.1050-1.el9.x86_64.rpm
```

12. If you want to start using the Linux VPN Client immediately during a 30-day trial period (see section 4.2 Trial period), restart the machine before you run the software in order to account for the users you just added.

The Linux VPN Client has been installed. You can use it for free during a 30-day trial period:

- To start a test tunnel, refer to chapter 8 Using the test tunnel.
- To install the **tgbtray** icon in the system menu, refer to chapter 5 tgbtray icon in the system menu.
- To configure VPN connections, refer to chapter 10 Configuring VPN connections.
- To learn about the client's control commands, refer to chapter 9 Command line.
- To activate the product, refer to chapter 4 Activating the software.
- To prevent the driver from starting with an unsupported kernel, refer to chapter 15 Selecting the kernel.

### 3.5.2 In Ubuntu

To install the Linux VPN Client from the command line in Ubuntu, proceed as follows:

1. If you downloaded the software package on a machine other than the one on which the Linux VPN Client is to be installed, copy it to the destination machine.
2. Open a terminal window (Ctrl + Alt + T).

3. Access the folder containing the `thegreenbow-vpn-client-3.4.r-b.ubuntu-24.04.deb` package (where **r** is the software revision number and **b** the software build number).
4. Run the following installation command:

```
sudo apt install ./ thegreenbow-vpn-client-3.4.r-b.ubuntu-24.04.deb
```

5. Add the VPN users to the `tgb` group by running the following command:

```
sudo usermod -aG tgb $(whoami)
```

6. Enter the administrator's password and press Enter.
7. If you want to start using the Linux VPN Client immediately during a 30-day trial period (see section 4.2 Trial period), restart the machine before you run the software in order to account for the users you just added.

The Linux VPN Client has been installed. You can use it for free during a 30-day trial period:

- To start a test tunnel, refer to chapter 8 Using the test tunnel.
- To install the **tgbtray** icon in the system menu, refer to chapter 5 tgbtray icon in the system menu.
- To configure VPN connections, refer to chapter 10 Configuring VPN connections.
- To learn about the client's control commands, refer to chapter 9 Command line.
- To activate the product, refer to chapter 4 Activating the software.
- To prevent the driver from starting with an unsupported kernel, refer to chapter 15 Selecting the kernel.

## 4 Activating the software

### 4.1 Introduction

You can use a fully functional version of the Linux VPN Client free of charge during a 30-day trial period (see section 4.2 Trial period below).

At the end of the 30-day trial period, you will no longer be able to use the software. If you want to continue using it, we will ask you to purchase a license.

Licenses are available on a subscription basis. Visit the Linux VPN Client page on [TheGreenBow](#)'s website for further details.

To activate the Linux VPN Client you must have superuser privileges (or root access) on the machine. You must also update the license file named `vpnsetup.json` as described below in section 4.4 Activation procedure.

### 4.2 Trial period

The `vpnsetup.json` license file that is installed by default contains 00000000000000000000000000 (24 zeros) in place of the license number and the email address of TheGreenBow's support.

This information is sufficient to use the software during the trial period. You do not need to make any changes to this file.

The number of days remaining before the trial period expires is specified whenever you run a `tgbdctl` command. When the trial period has expired, running the `tgbdctl` command will return the following error code:

```
-1 days
```



You will only be eligible for a trial period once.

### 4.3 Format and content of the `vpnsetup.json` file

The data to activate the Linux VPN Client must be entered into a text file named `vpnsetup.json` in ASCII format.

To do this, enter the license number you have received and the user's email address in an Activation section as follows:

```
{
  "license" : "123456789012345678901234",
  "email" : "username@company.com"
}
```

If you use a TAS activation server, you must also add the server's OSA parameters as follows:

```
{
  "license" : "123456789012345678901234",
  "email" : "username@company.com"
  "osaur1" : "192.168.217.102/osace_activation.php"
  "osaport" : "80"
  "osacert" : "MIICGjCCAYOgAwIBAgIBADANBg [.....]
muHf58kMO0jvhkyq24GryqptSaSJqVIA="
}
```



In the `osaur1` parameter, if the URL contains `https`, the protocol used will be `https`. Otherwise, the protocol used will be `http`.

## 4.4 Activation procedure

### 4.4.1 Automatic activation

To activate the Linux VPN Client, follow the steps described below:

1. Open a terminal window.
2. To update the `vpnsetup.json` license file, run the following command making sure to replace the **Xs** with the license number and `user@domain.com` with the email address associated with the license number:

```
echo -e "{\n\t\"license\" :
\n\"XXXXXXXXXXXXXXXXXXXXX\", \n\t\"email\" :
\n\"user@domain.com\" \n}" | sudo tee
/etc/tgb/vpnsetup.json
```

3. Run the following command to restart the service:

```
sudo systemctl restart tgbiked.service
```

4. Run the following command to display a log:

```
systemctl status tgbiked
```

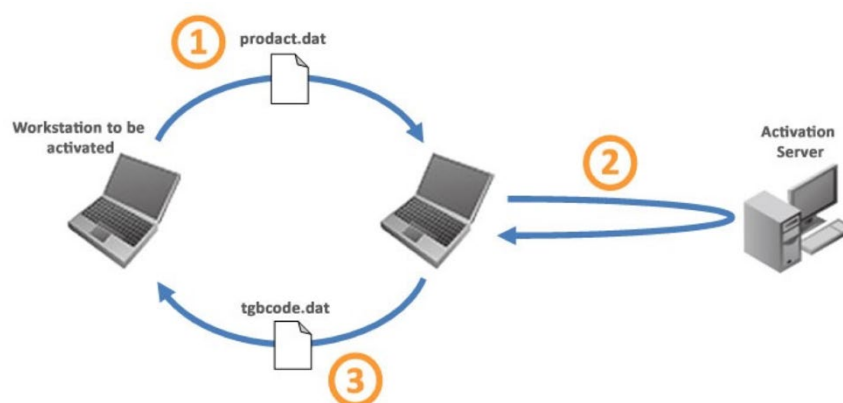
5. Check that the following message “Activation succeeded with license number 123456789012345678901234” is in the log.
6. If you haven't already done so after installing the software, restart the machine.



For information on how to display the log, refer to chapter 12 Logs.

#### 4.4.2 Manual activation

When activation fails because of a communication issue with the activation server, the software can be activated manually on [TheGreenBow's](#) website. The procedure is as follows:



##### ① `product.dat` file

Retrieve the `product.dat` file from the `/etc/tgb/.osa` directory on the workstation that you want to activate.<sup>1</sup>

##### ② Activation

On a workstation that is connected to the activation server<sup>2</sup>, open the manual activation page<sup>3</sup>, and post the `product.dat` file. Let the server automatically create the `tgbcode` before downloading it.

##### ③ `tgbcode` file

Copy the `tgbcode` file to the `/etc/tgb/.osa` directory on the workstation that you want to activate. Start the software; it will be activated.

<sup>1</sup> The `product.dat` file is a text file that contains the workstation information used for the activation. If this file cannot be found in the Documents directory, carry out the software activation steps on the workstation. This will generate the file even if activation fails.

<sup>2</sup> The activation server is TheGreenBow's server, which can be accessed on the internet.

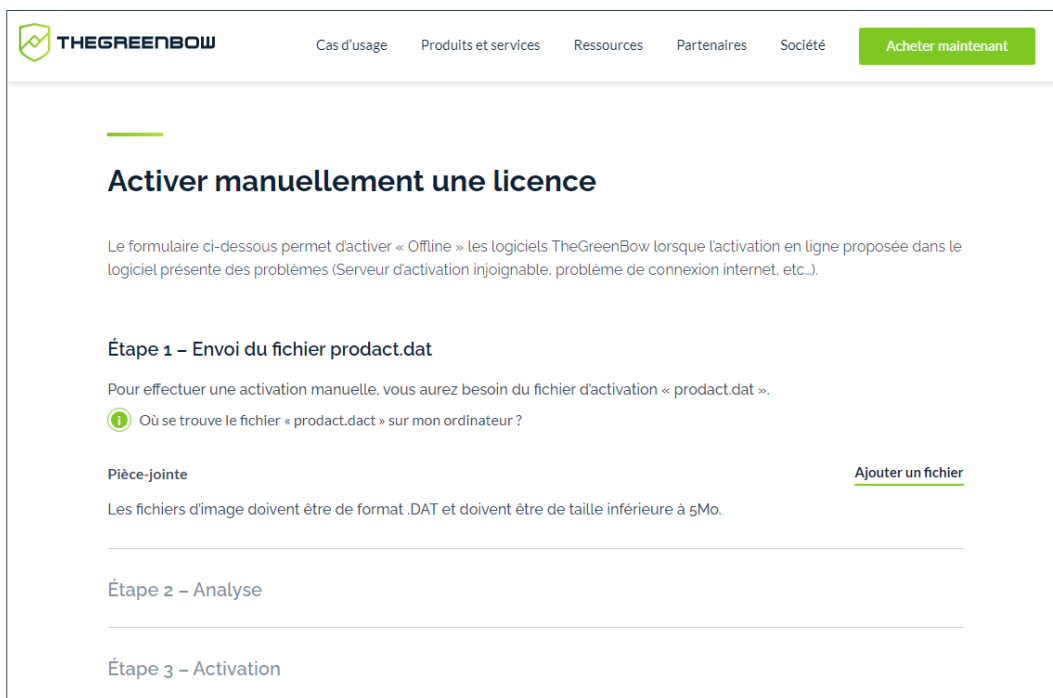
<sup>3</sup> Refer to the detailed procedure below.



To proceed with manual activation, follow the steps below:

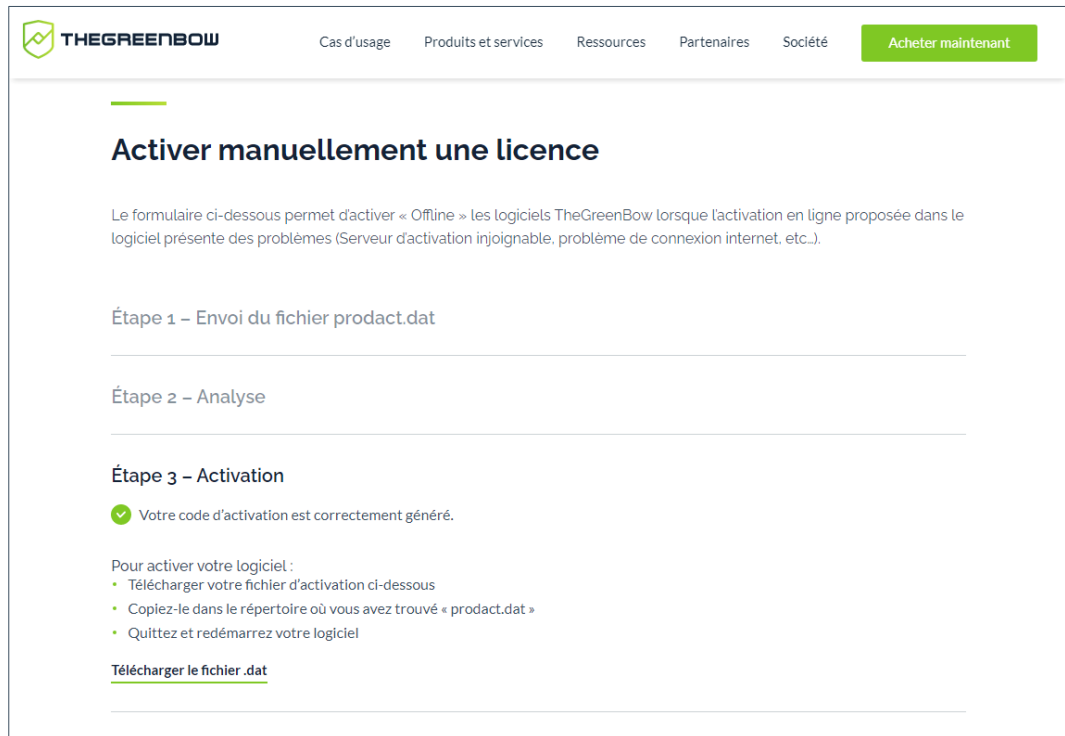
1. On a workstation connected to TheGreenBow's website, open the following webpage:

<https://www.thegreenbow.com/en/support/license-management/manual-license-activation/>



The screenshot shows the 'Activer manuellement une licence' page on the TheGreenBow website. The page has a green header with the logo and navigation links: 'Cas d'usage', 'Produits et services', 'Ressources', 'Partenaires', 'Société', and a green 'Acheter maintenant' button. The main content area has a green horizontal line above the title 'Activer manuellement une licence'. Below the title, a paragraph explains that the form allows activating 'Offline' software when online activation fails. The page is divided into three steps: 'Étape 1 - Envoi du fichier product.dat', 'Étape 2 - Analyse', and 'Étape 3 - Activation'. Step 1 includes instructions on needing a 'product.dat' file, a help icon with the question 'Où se trouve le fichier « product.dat » sur mon ordinateur ?', a 'Pièce-jointe' section with a file upload button 'Ajouter un fichier' and a note that files must be .DAT and under 5MB, and a text input field.

2. Click **Add a file** and open the `product.dat` file created on the workstation that you want to activate.
3. Click **Send**. The activation server will check the validity of the information contained in the `product.dat` file.
4. Click **Submit**. The activation server will provide a link to download a file containing the activation code for the workstation to be activated.



The screenshot shows the 'Activer manuellement une licence' page on the TheGreenBow website. The page has a navigation bar with links: 'Cas d'usage', 'Produits et services', 'Ressources', 'Partenaires', 'Société', and a green 'Acheter maintenant' button. The main heading is 'Activer manuellement une licence'. Below it, a paragraph explains that the form allows activating 'Offline' software when online activation fails. The process is divided into three steps: 'Étape 1 – Envoi du fichier product.dat', 'Étape 2 – Analyse', and 'Étape 3 – Activation'. Under 'Étape 3', a green checkmark indicates the activation code is generated. A list of instructions follows: download the activation file, copy it to the directory containing 'product.dat', and restart the software. A link 'Télécharger le fichier .dat' is provided.

The file name has the following format: `tgbcode_[date]_[code].dat` (e.g. `tgbcode__20231015_1029.dat`).

5. Copy the file you generated on our website to the `/etc/tgb/.osa` directory making sure to rename it as `tgbcode_0x19_1.dat`.

```
sudo cp tgbcod_YYMMDD_XXXX.dat
/etc/tgb/.osa/tgbcode_0x19_1.dat
```

6. Set the group that should have access to the file, i.e. group `tgb`, using the following CLI command:

```
sudo chown root:tgb /etc/tgb/.osa/tgbcode_0x19_1.dat
```

7. Restart the service using the following command:

```
sudo systemctl restart tgbiked.service
```



The above commands apply to both Red Hat and Ubuntu.



### 4.4.3 Activation successful

You are now ready to use the software. You can continue with the following steps:

- To start using the Linux VPN Client using a test tunnel, refer to chapter 8 Using the test tunnel
- To add an icon to the system menu, refer to chapter 5 tgbtray icon in the system menu.
- To create your VPN connection, refer to chapter 10 Configuring VPN connections

## 4.5 Activation errors

When activation has failed, the **tgbtray** icon can still be displayed in the system menu. In this case, an error message is displayed, and the icon turns orange.

If the log contains the message `Cancel starting UI Thread, product not activated` and/or `Activation failed: no activation parameters`, activation has failed. The Linux VPN Client stops immediately.

## 5 tgbtray icon in the system menu

The Linux VPN Client allows you to display an icon in the system menu (systray).

### 5.1 Adding the icon to the system menu



To add the icon in Red Hat, refer to section 5.1.1 In Red Hat.



To add the icon in Ubuntu, refer to section 5.1.2 In Ubuntu.

#### 5.1.1 In Red Hat

To add the **tgbtray** icon to the system menu in Red Hat, you must first start the default desktop environment GNOME<sup>1</sup>. Once this is done, follow the steps below:

1. Open a terminal window and run the following command:

```
sudo dnf install gnome-extensions-app gnome-shell-extension-appindicator
```

```
tgb@localhost:~ — sudo dnf install gnome-extensions-app gn...
[tgb@localhost ~]$ sudo dnf install gnome-extensions-app gnome-shell-extension-a
ppindicator
[sudo] password for tgb:
Updating Subscription Management repositories.
Last metadata expiration check: 1:55:01 ago on Mon 05 Feb 2024 08:16:24 PM CET.
Dependencies resolved.
=====
Package                Arch    Version      Repository      Size
=====
Installing:
  gnome-extensions-app  x86_64  40.0-3.el9   rhel-9-for-x86_64-appstream-rpms 125 k
  gnome-shell-extension-appindicator
                                noarch  49-1.el9     epel                                64 k
Transaction Summary
=====
Install 2 Packages

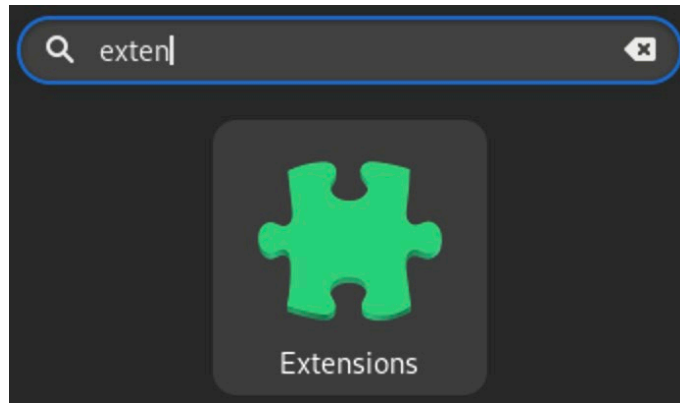
Total download size: 189 k
Installed size: 569 k
Is this ok [y/N]:
```

<sup>1</sup> If it is not installed, refer to the Red Hat documentation to find out how to do this.

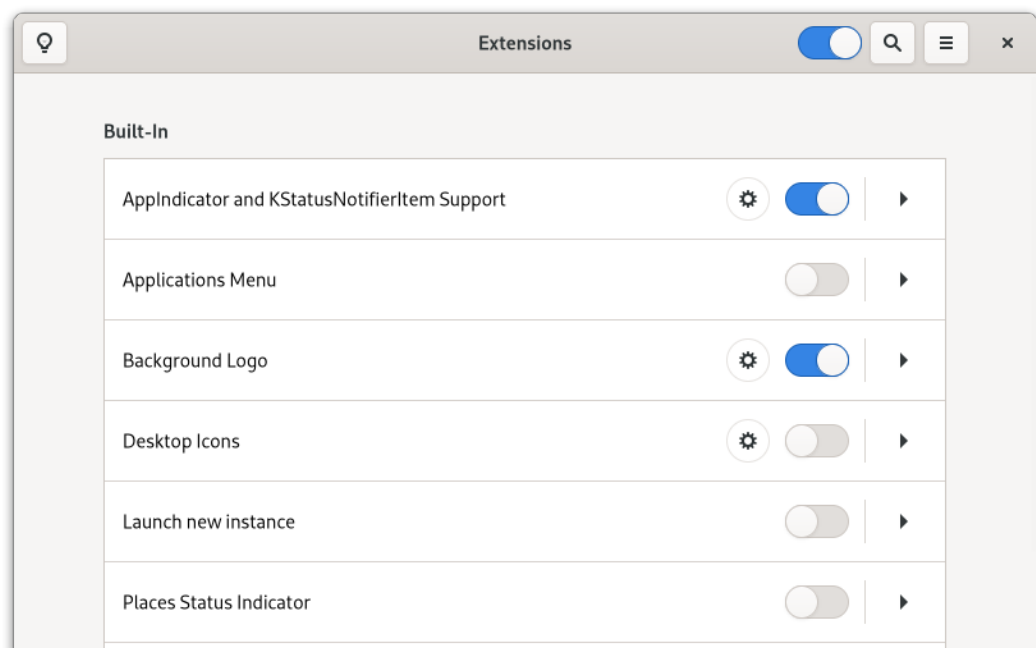
2. If you have not done so after installing the software, add the VPN user to the `tgb` group by running the following command:

```
sudo usermod -aG tgb $(whoami)
```

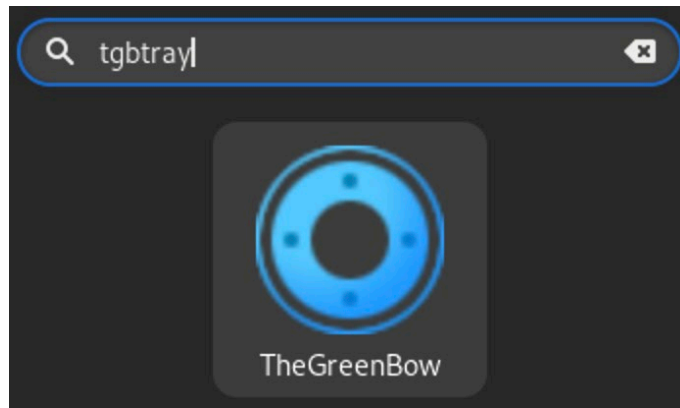
3. Restart the workstation.
4. Start the **Extensions** application, e.g. by running a search in the **Activity Overview**.



5. Enable **AppIndicator and KStatusNotifierItem Support**.



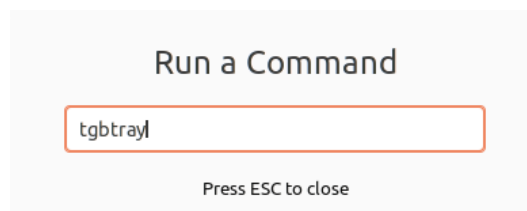
6. Start the **tgbtray** application, e.g. by running a search in the **Activity Overview**.



You will then be able to use the icon in the system menu.

### 5.1.2 In Ubuntu

To add the icon to the system menu in Ubuntu, click the TheGreenBow icon in the application list or press Alt + F2 to open the **Run a command dialog** and run the `tgbtray` command.



You can also add the icon by running the `tgbtray` command in a terminal window.

## 5.2 State of the system menu icon

The Linux VPN Client icon in the system menu changes color according to the status of the VPN connection:



Blue icon: no VPN connection is active.



Icon with spinning arrows: tunnel is being opened



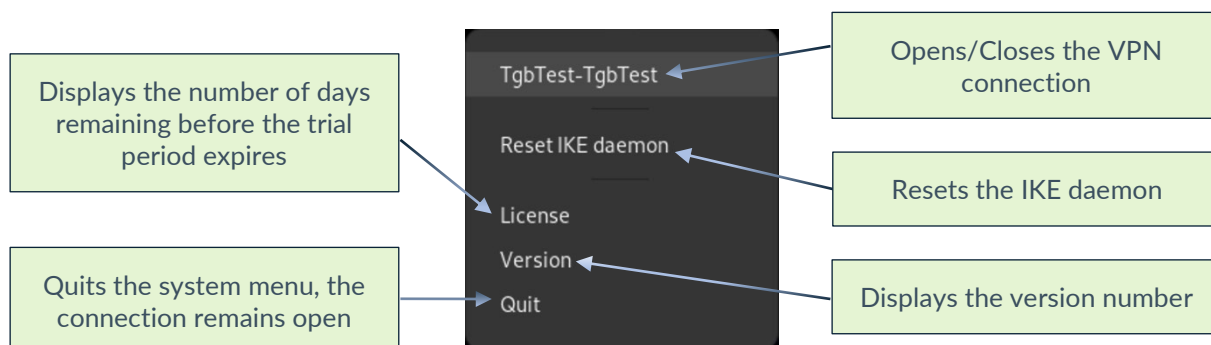
Orange icon: failed to open tunnel



Green icon: a VPN connection is active.

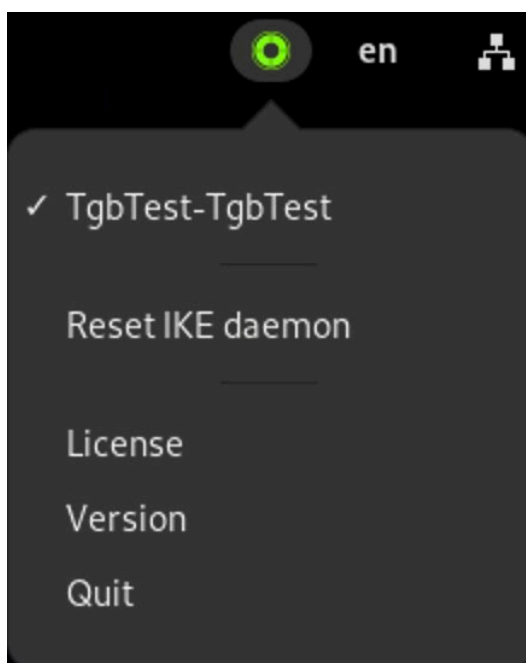
## 5.3 Contextual menu of the icon in the system menu

Click on the icon to expand the contextual menu. It contains the list of configured VPN connections as well as the following commands:



Select a VPN connection in the list to activate it. Only one connection can be active at a time. If you select another connection while a tunnel is open, the current tunnel is closed automatically before the new tunnel is opened.

The icon turns green as soon as a VPN connection is active, and a checkmark appears in front of the active VPN connection.



You can reset the **tgbray** icon in the system menu if it is no longer responding (see section 9.10 Resetting the tgbray icon).

## 6 Updating the software

If a previous version of the Linux VPN Client is already installed on your workstation, you can perform an update from the command line.



Before proceeding with an update, you must back up the `conf.tgb` configuration file and the `vpnsetup.json` license file (see section 6.1 Backing up below).



In version 2.0 of the Linux VPN Client, the activation file was named `vpnsetup.ini`. This file is no longer compatible with version 3.4. The `vpnsetup.json` file must now be used instead as described in the update procedure below.

### 6.1 Backing up before updating

To back up and restore the `conf.tgb` configuration file and the `vpnsetup.json` license file, proceed as follows:

1. Copy the `conf.tgb` configuration file and the `vpnsetup.json` license file located under `/etc/tgb/` to another secure folder.
2. Update the software as described in the next section.
3. Replace the `conf.tgb` file that has been installed with the one you backed up.
4. Copy the backed up `vpnsetup.json` license file to `/etc/tgb/`.



If you are performing an update from version 2.0 or lower of the software, you must generate the `vpnsetup.json` license file, as indicated in section 4.3 Format and content of the `vpnsetup.json` file.

### 6.2 Updating from the command line

Once you have backed up the configuration file (see section 6.1 Backing up before updating above), you can update the Linux VPN Client from the command line.



When updating to version 3.4.4 or higher, given that the machine identifier used to activate the machine has changed, you must delete the `/etc/tgb/.osa` folder and then restart the `tgbiked` service.



To update the Linux VPN Client from the command line in Red Hat, follow the steps below:

1. Make sure you have made a backup as described in section 6.1 Backing up before updating.
2. If you are performing an update to version 3.4.4, make sure you have deleted the `/etc/tgb/.osa` folder and restarted the `tgbiked` service, as described above.
3. Open a terminal window.
4. Navigate to the folder to which you have downloaded the installation package, e.g. `~/Downloads/`.
5. Proceed with the installation, as described in section 3.5 Installation procedure.
6. Copy the `conf.tgb` configuration file and the `vpnsetup.json` license file from the backup directory to `/etc/tgb/`.
7. Run the following command to restart the service:

```
sudo systemctl restart tgbiked.service
```

The Linux VPN Client has been updated. You can start using the software.



To avoid restarting the driver with an incompatible kernel, refer to chapter 15 Selecting the kernel.



If you are performing an update from version 2.0 or lower, we recommend uninstalling the software as described in chapter 7 Uninstalling the software, and then performing an installation as described in section 3.5 Installation procedure.

## 7 Uninstalling the software

When you no longer wish to use the Linux VPN Client, you can uninstall it from the command line.



For Red Hat, refer to section 7.1 In Red Hat.



For Ubuntu, refer to section 7.2 In Ubuntu.

### 7.1 In Red Hat

To uninstall the Linux VPN Client in Red Hat, proceed as follows:

1. Open a terminal window.
2. Run the following command:

```
sudo dnf remove thegreenbow-vpn-client.x86_64
```

This command deletes the files and dependent packages that were added during installation, and which are no longer used. Configuration files added later, e.g. `config.tgb`, will not be deleted.

```
tgb@localhost:~ — sudo dnf remove thegreenbow-vpn-client.x...
[tgb@localhost ~]$ sudo dnf remove thegreenbow-vpn-client.x86_64
[sudo] password for tgb:
Updating Subscription Management repositories.
Dependencies resolved.
=====
Package                Arch    Version      Repository      Size
=====
Removing:
thegreenbow-vpn-client x86_64 3.4.2-1.el9 @commandline      13 M
Removing unused dependencies:
pcsc-lite-libs          x86_64 1.9.4-1.el9 @rhel-9-for-x86_64-baseos-rpms 45 k
systemd-resolved        x86_64 252-18.el9  @rhel-9-for-x86_64-baseos-rpms 787 k

Transaction Summary
=====
Remove 3 Packages

Freed space: 14 M
Is this ok [y/N]:
```

The Linux VPN Client has been uninstalled.

### 7.2 In Ubuntu

To uninstall the Linux VPN Client in Ubuntu, proceed as follows:

1. Open a terminal window (Ctrl + Alt + T).

2. Run one of the following commands:

```
sudo apt remove thegreenbow-vpn-client
```

Or:

```
sudo apt purge thegreenbow-vpn-client
```

This command deletes the files added during installation as well as the configuration files added later, e.g. `config.tgb`, if it has been modified. Any other packages added during installation will not be deleted.



The difference between the `remove` command and the `purge` command is that the latter will let the system handle the deletion of all existing elements.

3. When appropriate, run the following command:

```
sudo apt autoremove
```

This command deletes the packages added during installation and that are no longer used.

The Linux VPN Client has been uninstalled.

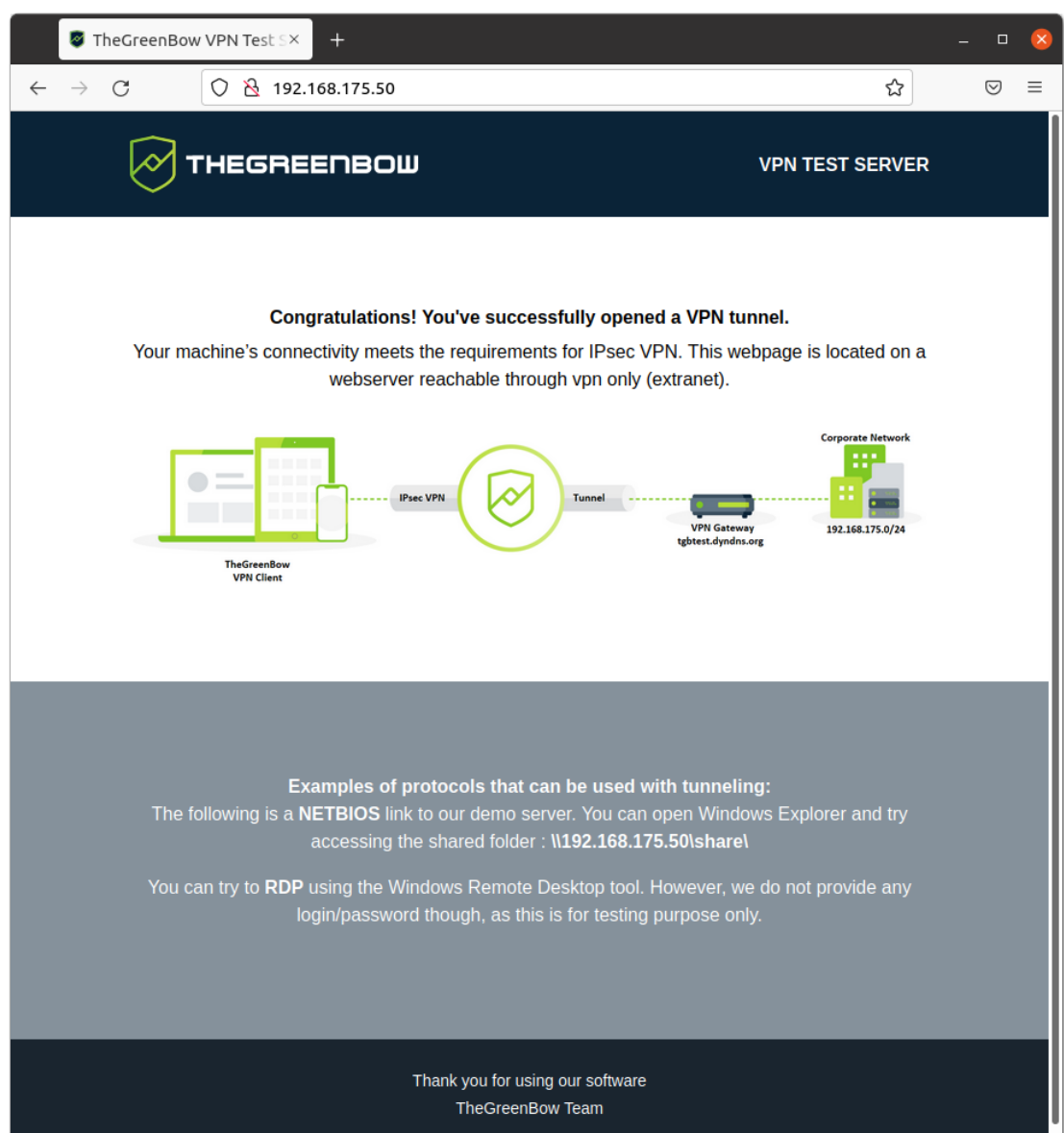
## 8 Using the test tunnel

A VPN configuration containing a test VPN tunnel called “TgbTest” is provided in the `conf.tgb` file located in the `/etc/tgb/` directory.

It is imported by default and allows you to test the Linux VPN Client by connecting to a test gateway.

The test configuration can be used to check whether the Linux VPN Client is operational.

Once the tunnel is open, you should be able to send a ping request to IP address 192.168.175.50 or open the <http://192.168.175.50/> web page in your browser.





## 9 Command line

### 9.1 Introduction

The Linux VPN Client provides a command line interface that enables you to carry out the following operations:

- Displaying help
- Displaying the version of the software
- Displaying the number of days remaining before the trial period expires
- Listing the configured VPN connections
- Opening a VPN connection
- Closing a VPN connection
- Displaying the status of the VPN connection



When you are using the Linux VPN Client without a license, the number of days remaining before the trial period expires is displayed each time you run a `tgbcctl` command.



The Linux VPN Client's control commands are identical regardless of the Linux distribution used.

### 9.2 Displaying help

To display the help, open a terminal window and run the following command:

```
tgbcctl --help
```

```

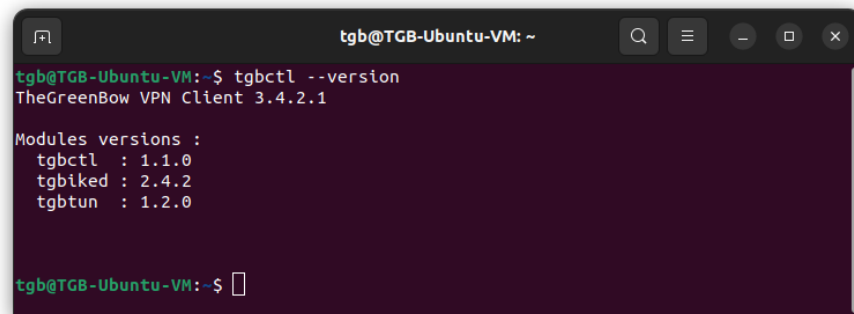
    tgb@TGB-Ubuntu-VM: ~
    tgb@TGB-Ubuntu-VM:~$ tgbcctl --help
    Usage : tgbcctl COMMAND

    Commands:
    --help          produce this help message
    --version       display tgbcctl version
    --licence       display licence remaining time
    list            display available tunnel
    reset           restart tgbiked
    status <tunnel> display tunnel state
    up <tunnel>      open tunnel
    down <tunnel>   close tunnel
    tgb@TGB-Ubuntu-VM:~$
```

## 9.3 Displaying the version of the software

To display the version of the software, open a terminal window and run the following command:

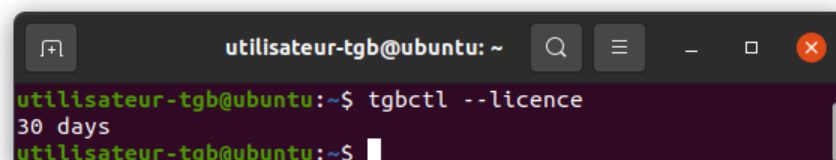
```
tgctl --version
```

A terminal window titled 'tgb@TGB-Ubuntu-VM: ~' showing the command 'tgb@TGB-Ubuntu-VM:~\$ tgctl --version' and its output. The output displays 'TheGreenBow VPN Client 3.4.2.1' followed by a table of module versions: 'Modules versions :', 'tgctl : 1.1.0', 'tgbiked : 2.4.2', and 'tgbtun : 1.2.0'. The prompt returns to 'tgb@TGB-Ubuntu-VM:~\$'.

## 9.4 Displaying the number of days remaining before the trial period expires

To display the number of days remaining before the trial period expires, open a terminal window and run the following command:

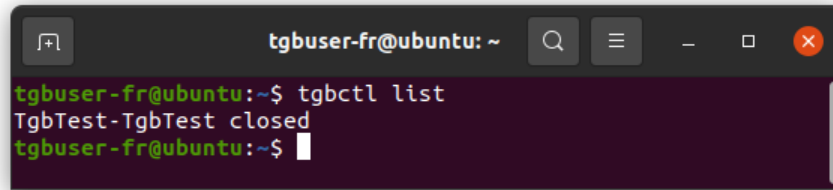
```
tgctl --licence
```

A terminal window titled 'utilisateur-tgb@ubuntu: ~' showing the command 'utilisateur-tgb@ubuntu:~\$ tgctl --licence' and its output. The output displays '30 days'. The prompt returns to 'utilisateur-tgb@ubuntu:~\$'.

## 9.5 Listing the configured VPN connections

To list the configured VPN connections, open a terminal window and run the following command:

```
tgctl list
```

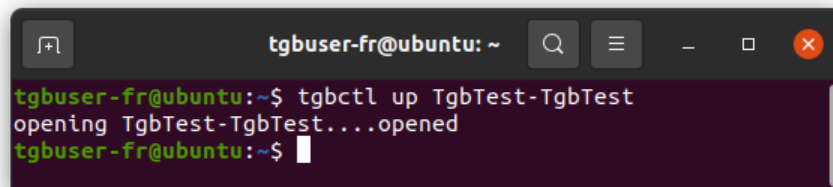


```
tgbuser-fr@ubuntu: ~  
tgbuser-fr@ubuntu:~$ tgbctl list  
TgbTest-TgbTest closed  
tgbuser-fr@ubuntu:~$
```

## 9.6 Opening a VPN connection

To open a VPN connection, open a terminal window and run the following command:

```
tgbctl up [connection_name]
```

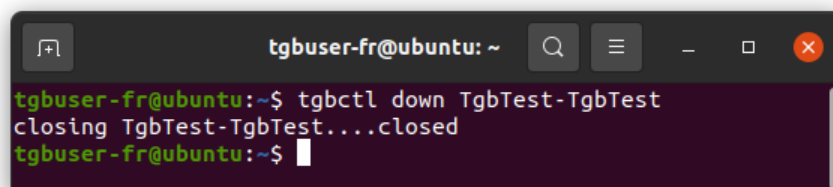


```
tgbuser-fr@ubuntu: ~  
tgbuser-fr@ubuntu:~$ tgbctl up TgbTest-TgbTest  
opening TgbTest-TgbTest....opened  
tgbuser-fr@ubuntu:~$
```

## 9.7 Closing a VPN connection

To close a VPN connection, open a terminal window and run the following command:

```
tgbctl down [connection_name]
```

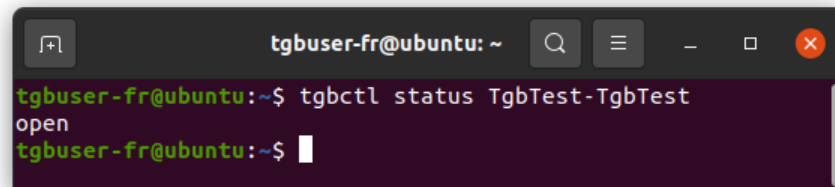


```
tgbuser-fr@ubuntu: ~  
tgbuser-fr@ubuntu:~$ tgbctl down TgbTest-TgbTest  
closing TgbTest-TgbTest....closed  
tgbuser-fr@ubuntu:~$
```

## 9.8 Displaying the status of the VPN connection

To display the status of a VPN connection, open a terminal window and run the following command:

```
tgbdctl status [connection_name]
```



## 9.9 Setting a PIN code

For tunnel automation purposes, the `--pincode` argument can be used to enter a smart card PIN code directly from the command line:

```
tgbdctl up [tunnel_name] --pincode [PIN_code]
```

## 9.10 Resetting the tgbtray icon

To reset the **tgbtray** icon in the system menu, open a terminal window and run the following command:

```
tgbtray reset
```

Run this command when you want to force reload the icon in the system menu.



Any user can run this command without needing any administrator rights.



## 9.11 Resetting the IKE daemon

To reset the IKE daemon, open a terminal window and run the following command:

```
tgbdctl reset
```

Run this command when the tunnel goes down and you are unable to restart it.



Any user can run this command without needing any administrator rights.

## 10 Configuring VPN connections

### 10.1 Introduction

TheGreenBow VPN clients rely on a VPN configuration that defines the list of VPN connections that the administrator makes available to the workstation user. This file, also called a configuration file, must be named `conf.tgb`.

The Linux VPN Client does not provide an HMI to build or modify the VPN configuration.

This feature is available in our Windows Enterprise VPN Client v7.4.



If you are running Windows Enterprise VPN Client version v7.5 or higher, you will need to manually update the configuration file to ensure compatibility with the Linux VPN Client.



Refer to the Windows Enterprise VPN Client's Deployment Guide (see chapter 18 Related reference documents).

If you are an administrator, you must use the Windows Enterprise VPN Client to generate a VPN configuration as specified in section 10.5 Updating the VPN configuration.



If you want to manage several configuration files, rename unused ones, for example, to `conf.tgb.bak`.

### 10.2 Using IPsec DR mode

To be able to run the Linux VPN Client in IPsec DR mode, the configuration must include the following dynamic parameters:

- `NoNATTNegotiation` set to the value `true`
- `nonce_size` set to the value `16`
- `sha2_in_cert_req` set to the value `true`



Refer to the « Configuration Guide » entitled *Client VPN et IPsec DR* (available in French only) for further details about how to configure this mode.

## 10.3 Protecting the VPN configuration

The Linux VPN Client relies on the Linux operating system to protect the configuration. The configuration file is only accessible to users with superuser privileges.

No other user can modify the configuration file or inject a new one, which guarantees its authenticity and integrity.

The VPN configuration file is stored in the `conf.tgb` file under the `/etc/tgb/` directory.

The rights to this file are `-rw-----`, owner `root`. A standard user therefore cannot gain read or write access to the VPN configuration.

## 10.4 Managing certificates

The Linux VPN Client includes a selection of interfacing functions with all types of certificates, issued by any PKI, and on various types of storage devices, such as smart cards, tokens, and configuration files.

More specifically, the Linux VPN Client implements the following features:

- PKCS #11 access to tokens and smart cards
- Selection of certificates to use according to multiple criteria: subject, key usage, etc.
- Management of user certificates (VPN client end), such as VPN gateway certificates, including verification of validity dates, certificate chains, as well as root and intermediate certificates
- Certificate authority (CA) management

The certificates to be used are configured and specified in the Windows Enterprise VPN Client.



Refer to the Windows Enterprise VPN Client's Deployment Guide (see chapter 18 Related reference documents).

To use a certificate, proceed as follows:

1. In the Windows Enterprise VPN Client, import the user certificate and the associated CAs into your configuration (refer to the Windows Enterprise VPN Client "Administrator's Guide").
2. Follow the procedure for updating the configuration described in section 10.5 Updating the VPN configuration below.

The certificate has been imported to the Linux VPN Client's configuration.

## 10.5 Updating the VPN configuration

To modify the configuration of your Linux VPN Client, proceed as follows:

1. Generate the configuration using the Windows Enterprise VPN Client.
2. Export the configuration in TGB format, without any password-protection, and name it `conf.tgb`.
3. Replace the `conf.tgb` file in the `/etc/tgb/` directory on the machine on which you want to import the configuration.
4. Run the following command to restart the service:

```
sudo systemctl restart tgbiked.service
```

You have updated the Linux VPN Client's VPN configuration.



Do not use the `tgbctl reset` or `tgbtray reset` command to load the configuration after updating it.



## 11 Using tokens and smart cards

### 11.1 Introduction

The Linux VPN Client now allows users to authenticate using a token or smart card. To enable this feature, you must carry out the following:

- Configure the virtual machine, where appropriate
- Install the token or smart card manufacturer's middleware, or a compatible middleware
- Generate the `vpnconf.ini` file that enables the VPN client to use the token or smart card

The Linux VPN Client supports a great number of tokens and smart cards that can be used for strong multi-factor authentication (MFA) using the PKCS #11 API.

PKCS #11 is an API to access cryptographic tokens and smart cards that has been standardized by RSA Labs. Most tokens and smart cards are compatible with PKCS #11. For the Linux VPN Client to be able to use the PKCS #11 API, a middleware provided by the manufacturer of the token or smart card must first be installed on the target computer.

The tokens and smart cards compatible with the Linux VPN Client are the ones listed on our website at

<https://www.thegreenbow.com/en/support/integration-guides/compatible-vpn-tokens/> and for which the item PKC has a green dot in front of it and is followed by the entry "Tested and qualified".



In principle, any token or smart card for which there is a PKCS #11 middleware can be used with the Linux VPN Client.

To be able to use tokens or smart cards with the Linux VPN Client, you must specify their characteristics in a PKCS #11 initialization file named `vpnconf.ini`, as described below.

### 11.2 `vpnconf.ini` file

To enable the Linux VPN Client to support tokens or smart cards that are not recognized as standard, you must create a `vpnconf.ini` file in the VPN Client's installation directory (`/etc/tgb/` by default). You can create the file using a standard text editor (e.g. nano).

The parameters to be specified in the `vpnconf.ini` file are grouped in a series of `ATR` sections used to define the attributes of tokens or smart cards that are not recognized as standard by the software.

ATR stands for “Answer To Reset”. It is an identifier that the token or smart card returns upon receiving a reset command. This identifier is related to the manufacturer and model of the token or smart card.

Each ATR section describes the required characteristics to access a token or smart card, or a family of tokens or smart cards that are not yet known to the software.

The parameters to be specified in the ATR section are detailed in the following table:

Parameter	Meaning
[ATR#]	ATR of the token or smart card to be added
mask	Mask to be used with this ATR <sup>1</sup>
scname	Name of the token or smart card (strictly descriptive field)
manufacturer	Name of the manufacturer (strictly descriptive field)
pkcs11dllname	Name of the PKCS #11 shared library (strictly descriptive field)
dllpath	Path to the PKCS #11 shared library. The path is the complete path. It must also contain the name of the shared library. <sup>2</sup>



Proceed with caution when entering the path to the PKCS #11 shared library in the `dllpath` parameter. If the path is not entered correctly, it may cause undesirable behavior in the software.



To retrieve information about a token connected to the workstation, you can use the `pcsc_scan` command (available with the `pcsc-tools` package).

<sup>1</sup> Details regarding ATRs and ATR masks are provided by the manufacturers of tokens or smart cards. If in doubt, you can configure a mask containing only FF. The lengths of the ATR and the mask must be identical. The `mask` line can thus be as follows:

```
mask=FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF
```

<sup>2</sup> The `dllpath` parameter must be defined.



## Example

```
[3B:0F:52:4E:42:4F:24:00:23:00:00:00:00:00:00:01]
mask="FF:FF:FF:FF:FF:FF:FF:00:FF:00:00:FF:FF:00:00:FF"
sname="Card Name"
manufacturer="Company Name"
pkcs11dllname="mdlw.so"
dllpath="/usr/lib/mdlw.so"
```

## 11.3 Installing the middleware

Refer to the middleware editor's documentation for installation instructions.

### Example with OpenSC in Ubuntu

OpenSC is an open source middleware that supports various tokens and smart cards.

To install the OpenSC middleware in Ubuntu, follow the steps below:

1. Open a terminal window (Ctrl + Alt + T).
2. Successively run the following commands:

```
sudo apt-get update -y
sudo apt-get install -y opensc
sudo apt-get install -y opensc-pkcs11
```

Once the OpenSC middleware has been installed, you must set the `pin_cache_ignore_user_consent` to true in the `opensc.conf` file.

Example for `/etc/opensc.conf`:

```
app default {
    # debug = 3;
    # debug_file = opensc-debug.txt;
    framework pkcs15 {
        pin_cache_ignore_user_consent = true;
    }
}
```

You can then proceed with creating the `vpnconf.ini` file.

## 11.4 Creating the vpnconf.ini file

To be able to use the Linux VPN Client with a token or smart card, you must carry out the following steps:

- Create the `vpnconf.ini` file using a text editor
- Add the information concerning the token or smart card
- Place it in directory `/etc/tgb/`

### Example for the Yubikey 5 NFC token

The following information must be entered in the `vpnconf.ini` file for a Yubikey 5 NFC token:

```
[3B:FD:13:00:00:81:31:FE:15:80:73:C0:21:C0:57:59:75:62:69:4B:65:79:40]
mask="FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF:FF"
sname="Yubikey 5 NFC"
manufacturer="Yubico"
pkcs11DllName="yubikey"
dllpath="/usr/lib/x86_64-linux-gnu/opensc-pkcs11.so"
```

You can now use the Yubikey 5 NFC token for authentication with the Linux VPN Client.





## 12 Logs

### 12.1 Introduction

The logs of the IKE daemon are stored using the log management provided by `systemd`.

To display the logs of the IKE daemon, run the following command in a terminal window:

```
journalctl -t tgbiked
```

### 12.2 Exporting in text format

To export the contents of the log in text format, run the following command in a terminal window:

```
journalctl -t tgbiked > [my_log_file.log]
```

Customer support is based on this file.

Should the support team ask you for the log file, make sure to also provide the following details in order for the support staff to have all the information it requires at hand:

- Version of the binary package used
- Version of the Linux distribution
- Version of the Linux kernel
- Version of the GNU C library (glibc)

To get information concerning the distribution and kernel, run the following command in a terminal window:

```
uname -a
```

To get information concerning the `glibc` library, run the following command in a terminal window:

```
ldd --version
```

## 12.3 Defining the log level

The log level can be defined in the following file:  
`/lib/systemd/system/tgbiked.service`.

The default log level is 6 `info`.

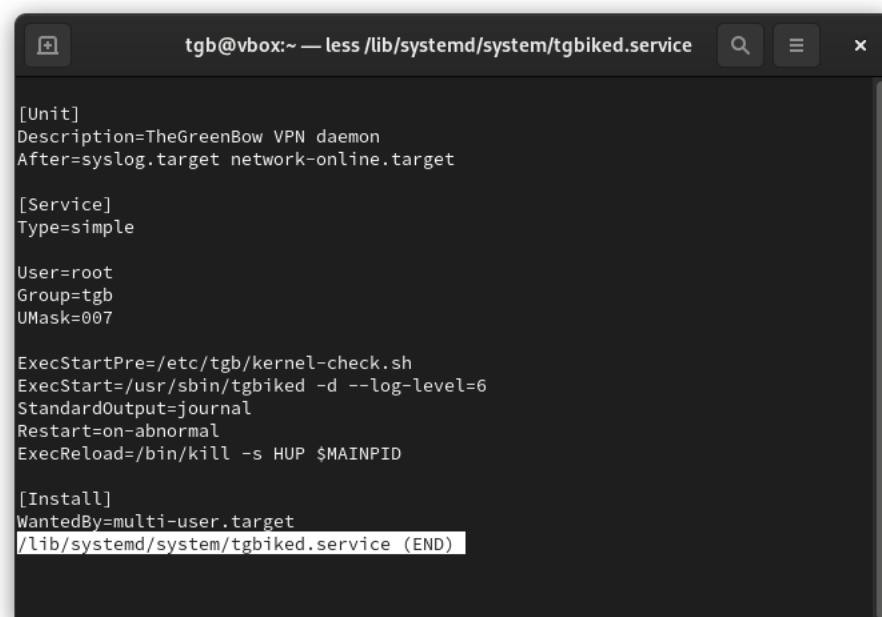
### 12.3.1 Displaying the current log level

To display the current log level, proceed as follows:

1. Open a terminal window and run the following command:

```
less /lib/systemd/system/tgbiked.service
```

The content of the file is displayed:



```
tgb@vbox:~ — less /lib/systemd/system/tgbiked.service

[Unit]
Description=TheGreenBow VPN daemon
After=syslog.target network-online.target

[Service]
Type=simple

User=root
Group=tgb
UMask=007

ExecStartPre=/etc/tgb/kernel-check.sh
ExecStart=/usr/sbin/tgbiked -d --log-level=6
StandardOutput=journal
Restart=on-abnormal
ExecReload=/bin/kill -s HUP $MAINPID

[Install]
WantedBy=multi-user.target
/lib/systemd/system/tgbiked.service (END)
```

2. Search for the following line and check the value of the `log-level` parameter:

```
ExecStart=/usr/sbin/tgbiked -d --log-level=6
```

3. Press `q` to quit `less` and return to the shell prompt.



### 12.3.2 Chaging the log level

To change the log level, proceed as follows:

1. Open a terminal window.
2. Open the file `tgbiked.service` in an editor of your choice, e.g. `nano`:

```
sudo nano /lib/systemd/system/tgbiked.service
```

3. Change the log level as required. For example, to enable level 7 debug, change the following line:

```
ExecStart=/usr/sbin/tgbiked -d --log-level=7
```

4. Save the file and quit the editor. Forr example, in `nano`, press `Ctrl + X`.
5. Confirm that you want to save the file and overwrite the existing one.

## 13 Configuring NetworkManager

To be able to run scripts when opening and closing a tunnel, you must use the NetworkManager daemon, which is available both in Red Hat and in Ubuntu.

To add scripts, follow the steps below:

1. Start by adding the `tgbtun0` virtual network interface to the interfaces managed by NetworkManager. To do this, add the following lines to the NetworkManager configuration file `/etc/NetworkManager.conf`:

```
[device]
match-device=interface-name:tgbtun0
managed=1
```

2. Add a script in `/etc/NetworkManager/dispatcher.d` to capture the opening (up) and closing (down) events of the tunnel associated with `tgbtun0`:

```
#!/usr/bin/bash
INTERFACE="$1"
EVENT="$2"

if [[ "$INTERFACE" == "tgbtun0" ]]
then
    if [[ "$EVENT" == "up" ]]
    then
        echo "tgbtun0 is up" >> /tmp/tgbtun.log
    fi
    if [[ "$EVENT" == "down" ]]
    then
        echo "tgbtun0 is down" >> /tmp/tgbtun.log
    fi
fi
```

The `tgbtun0` network interface is enabled (up) when the tunnel is open and disabled (down) when the tunnel is closed.



For more information, refer to the NetworkManager guide by running the `man NetworkManager-dispatcher` command.



## 14 Automatic startup

This chapter described the procedures to enable the following automation:

- Start `tgbray` automatically
- Open the tunnel automatically

### 14.1 Starting `tgbray` automatically

You can set the Linux VPN Client to launch automatically when the GUI starts. This is performed in the **Startup Applications**. The procedure is fairly similar for Red Hat and Ubuntu.



For Red Hat, refer to section 14.1.1 In Red Hat.

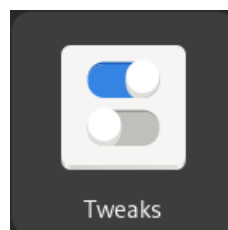


For Ubuntu, refer to section 14.1.2 In Ubuntu.

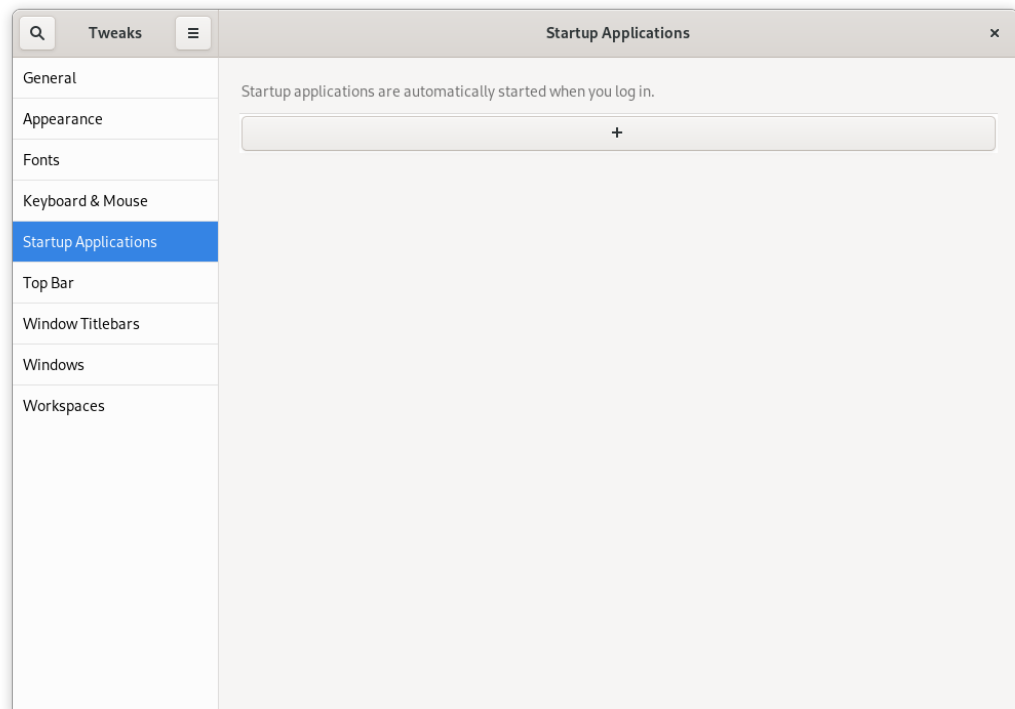
#### 14.1.1 In Red Hat

To add the `tgbray` icon to the startup applications, follow the steps below:

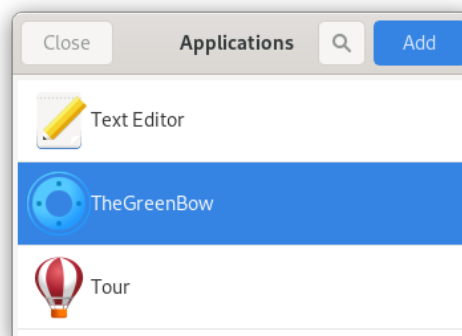
1. If you have not already done so install `gnome-tweaks`.
2. Open **Tweaks**.



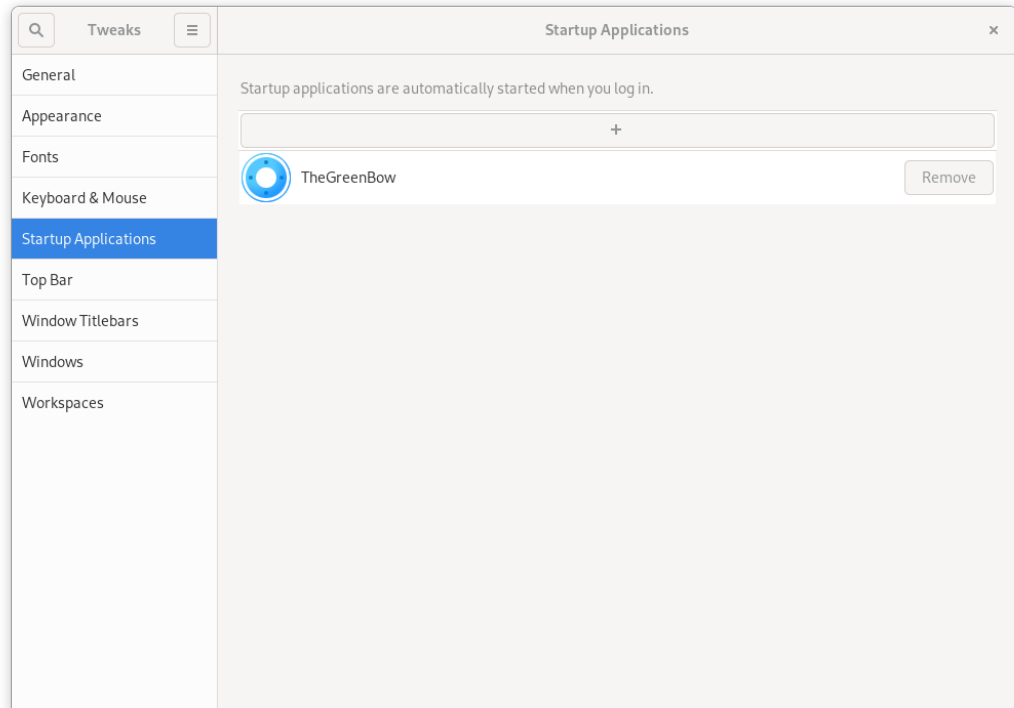
3. Select **Startup Applications** in the left side bar.



4. Click the plus sign button (+). A list of applications is displayed:



5. Select **TheGreenBow** from the list of available applications and click **Add**. The application is added to the applications that are automatically started when you log in:



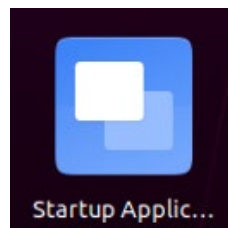
6. Close the window.

The next time you log in to your user account, the **tgbray** icon will automatically be loaded in the system menu.

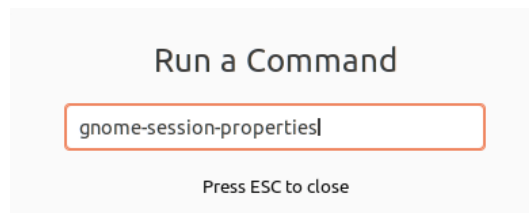
### 14.1.2 In Ubuntu

To add the **tgbray** icon to the startup applications, follow the steps below:

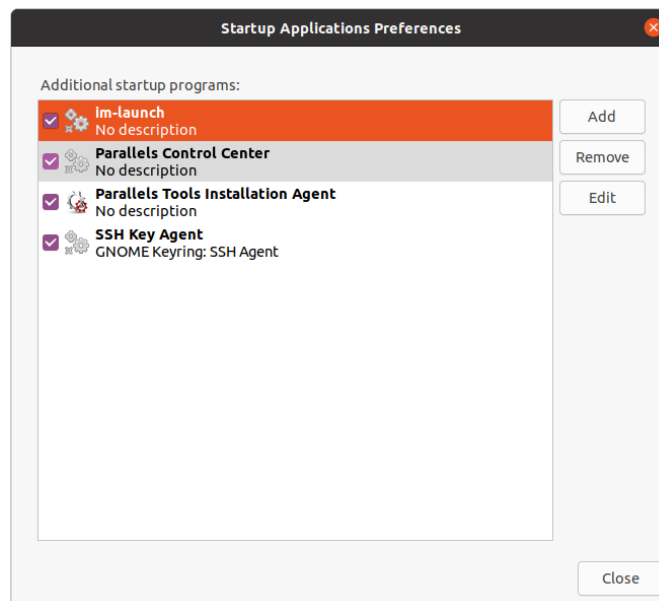
1. Open the application list by clicking the **Show Applications** button in the lower-left corner of your screen, and then click the **Startup Applications Preferences** icon.



Alternative: Press **Alt + F2** to open the **Run a Command** dialog and run the `gnome-session-properties` command.

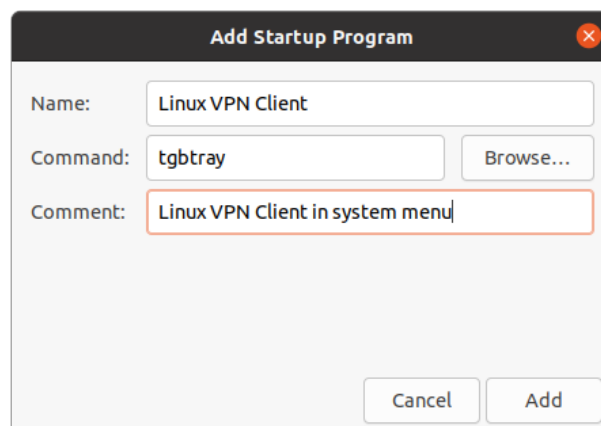


The **Startup Applications Preferences** window is displayed.



2. Click **Add**.

The **Add Startup Program** window is displayed.



3. Enter a name, e.g. **Linux VPN Client**.
4. Enter the `tgbtray` command.
5. Add a comment, e.g. **Linux VPN Client** icon in system menu.
6. Click **Add**.

The next time you log in, the Linux VPN Client will start automatically and its icon will be shown in the system menu.





## 14.2 Opening the tunnel automatically



Before you can enable automatic tunnel opening, you must first configure automatic startup for `tgbray` (see section 14.1 Starting `tgbray` automatically).



The procedure described below applies both to Red Hat and Ubuntu.

To open the tunnel automatically, follow the steps below:

1. If `tgbray` is already running, quit it.
2. Add the following lines to the `/etc/tgb/conf.tgb` configuration file:

```
<cfg_automation>
  <autostart>true</autostart>
</cfg_automation>
```

3. Run the following command to reset the IKE daemon:

```
tgbrctl reset
```

4. Run the following command to reset `tgbray`:

```
tgbray reset
```



See chapter 13 Configuring NetworkManager for more information about adding scripts.

## 15 Selecting the kernel

In Linux VPN Client versions 3.4.1 and earlier, the `tgbtun` driver was started automatically at system startup. To prevent the driver from starting with an incompatible kernel version and potentially leading the system to become unresponsive, as of version 3.4.4 of the Linux VPN Client, the driver is started before the `tgbiked` service is started and after checking the kernel version.

To anticipate kernel compatibility issues when the workstation's operating system is updated with a kernel that is not compatible with the installed version of the Client VPN Linux, you can specify the desired kernel in the following file: `/etc/tgb/tgbtun.params`. This will for instance prevent the service from being started in Ubuntu 22.04 with a version 6.x kernel.

The file `/etc/tgb/tgbtun.params` must only contain a single line starting with `requires` followed by the kernel number in the form of a regular expression, e.g.:

```
requires=5.14.*.el9_4.x86_64
```



## **16**      **Current limitations**

The current version of the Linux VPN Client has the following limitations:

- No encrypted configurations can be imported.
- Only a single VPN connection can be open at a time.

## 17 Managing errors

### 17.1 User must belong to "tgb" group

If you have not added the current user to the `tgb` user group, the following error message is displayed when you run commands:

```
ERROR: User must belong to "tgb" group
```

To add the user to the `tgb` group, open a terminal window and run the following command:

```
sudo usermod -aG tgb $(whoami)
```

You must log out and log back in for Ubuntu to take into account this command. In some cases, you even need to restart the system. We recommend that you restart the system in all cases.

### 17.2 Cannot get VPN connection list

When running the `tgbctl up [connection_name]` command, the following error may be displayed:

```
Error: Can't get tunnel list, check if tgbiked service  
is started  
can't be open: check status
```

To check whether the user has been added to the `tgb` group, open a terminal window and run the following command:

```
id
```

If the `tgb` group is not in the list, restart the machine in order for the group you've added to be taken into account.

### 17.3 Opening a VPN connection failed

When the Linux VPN Client fails to open a VPN connection, the following error message is displayed:

```
Opening [connection_name] ..... failed
```

When opening a VPN connection has failed, open a terminal window and run the following command:

```
journalctl -r -t tgbiked
```

You can analyze the log yourself (see chapter 12 Logs) or contact the support team: <https://www.thegreenbow.com/en/support/online-support/technical-support/>.

## 17.4 Non-root users must not be able to access the configuration file

When the Linux VPN Client is unable to open a VPN connection after having replaced the `/etc/tgb/conf.tgb` file with a new configuration, check the log to see whether it contains the following message: "Non-root users must not be able have access to file `/etc/tgb/conf.tgb`". Users other than superusers should not be able to access the configuration file.

If this is the case, run the following command:

```
sudo chmod 600 /etc/tgb/conf.tgb
```

## 17.5 Checking the driver

Run the following command to check whether the driver is loaded:

```
lsmod | grep tgb
```

The command must return the following message:

```
tgbtun          [ID]  0
```

If this is not the case, follow the steps below:

1. Check that the driver `tgbtun.ko.xz` is available in the following folder :
  - o `/lib/modules/`uname -r`/extra/1` for Red Hat ;
  - o `/lib/modules/`uname -r`/updates/dkms/2` for Ubuntu.
2. Check the directory `/usr/src/tgbtun-1.2` exists.

---

<sup>1</sup> On some systems, it is preferable to specify `/usr/lib/modules`.

<sup>2</sup> *ibid*

3. If the driver is not installed, refer to the next section to install it.

If the issue remains, contact the support team with the results from the following commands:

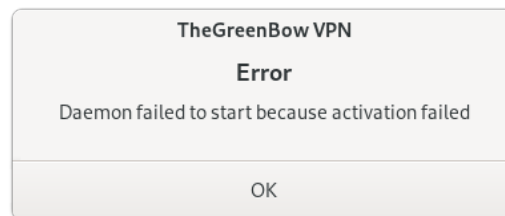
- `modinfo tgbtun`
- `dkms status tgbtun`

Please use the following form to contact the support team:

<https://www.thegreenbow.com/en/support/online-support/technical-support/>.

## 17.6 Cannot start IKE daemon

If the following error is returned when you try starting the software after installation:



Follow the steps below to check whether the driver is loaded and, where required, install it manually:

1. Check the driver as described in section 17.5 Checking the driver.
2. If the command does not return anything, run the following commands successively to install the driver:

```
cd /usr/src/tgbtun-1.2
sudo dkms install tgbtun/1.2
```

3. In Ubuntu, run the following command:

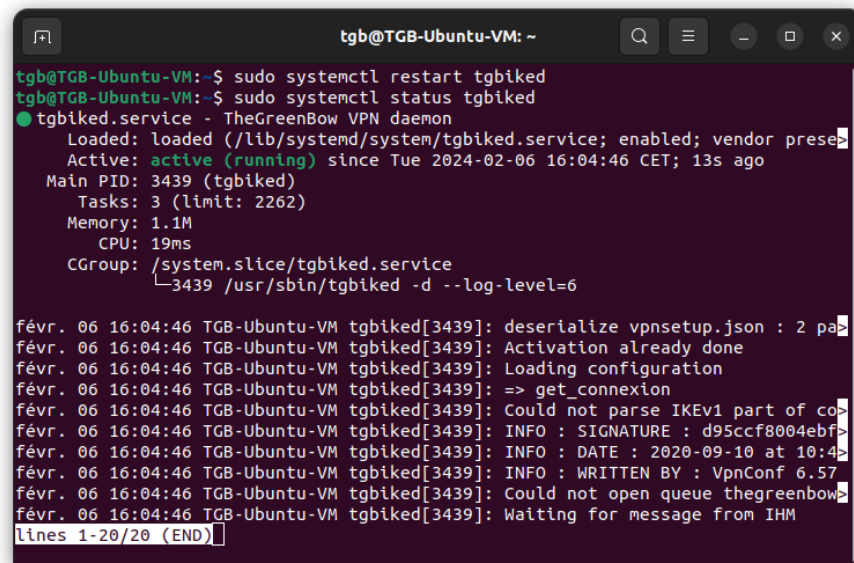
```
sudo insmod /lib/modules/`uname -r`/updates/dkms/tgbtun.ko
```

In Red Hat, jump directly to step 4.

4. Now, run the following commands successively to restart the IKE daemon and display its status:

```
sudo systemctl restart tgbiked
sudo systemctl status tgbiked
```

Information similar to what's shown in the screenshot below should be returned:



```
tgb@TGB-Ubuntu-VM: ~  
tgb@TGB-Ubuntu-VM:~$ sudo systemctl restart tgbiked  
tgb@TGB-Ubuntu-VM:~$ sudo systemctl status tgbiked  
● tgbiked.service - TheGreenBow VPN daemon  
   Loaded: loaded (/lib/systemd/system/tgbiked.service; enabled; vendor prese  
   Active: active (running) since Tue 2024-02-06 16:04:46 CET; 13s ago  
     Main PID: 3439 (tgbiked)  
       Tasks: 3 (limit: 2262)  
      Memory: 1.1M  
         CPU: 19ms  
     CGroup: /system.slice/tgbiked.service  
             └─3439 /usr/sbin/tgbiked -d --log-level=6  
  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: deserialize vpnsetup.json : 2 pa  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: Activation already done  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: Loading configuration  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: => get_connexion  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: Could not parse IKEv1 part of co  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: INFO : SIGNATURE : d95ccf8004ebf>  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: INFO : DATE : 2020-09-10 at 10:4>  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: INFO : WRITTEN BY : VpnConf 6.57  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: Could not open queue thegreenbow>  
févr. 06 16:04:46 TGB-Ubuntu-VM tgbiked[3439]: Waiting for message from IHM  
lines 1-20/20 (END)
```

You can then finish installation by adding the **tgbtray** icon to the system menu (see chapter 5 tgbtray icon in the system menu), and then open a test tunnel (see chapter 8 Using the test tunnel).

If you still cannot start the IKE daemon, please contact the support team to understand what happened:

<https://www.thegreenbow.com/en/support/online-support/technical-support/>.

## 17.7 IKE daemon is unresponsive

If the IKE daemon becomes unresponsive, which may happen after a network interface change or after disconnecting and reconnecting the network cable, run the following command to restart it:

```
sudo kill -9 $(pidof tgbiked)
```

## 17.8 Token or smart card errors

When a user enters an incorrect PIN code, the HMI used to request VPN connection to be opened informs the user that the PIN code entered is not valid.

When the user locks the token or smart card by entering an incorrect PIN code several times, the HMI informs the user that the token or smart card is locked.

In addition, the Linux VPN Client may return the following errors related to the use of tokens and smart cards:

- `no smartcard plugged`: the token or smart card is not connected
- `wrong pin code`: the PIN code entered by the user is incorrect
- `smartcard is locked`: the token or smart card is locked

## 17.9 Virtual machine does not recognize a token or smart card

If you are running the operating system in a VMware virtual machine, start by following the procedure below:

1. Stop the virtual machine.
2. Locate the `*.vmx` configuration file of the virtual machine (refer to this [VMware knowledge base article](#) that addresses this topic).
3. Open the configuration file in a text editor.
4. Add the following two lines to the file and save it:

```
usb.generic.allowHID = "TRUE"
usb.generic.allowLastHID = "TRUE"
```

The virtual machine is now ready to use the token or smart card. You can now proceed with installing the middleware.



If you are using another virtualization software, the principle remains the same: make sure that the virtual machine can access the token or smart card via USB.





## 18 Related reference documents

To find out how to generate the configuration file to be used with the Linux VPN Client, please refer to the Client VPN Windows Enterprise “Administrator’s Guide”. You will find it on [TheGreenBow](https://www.thegreenbow.com/en/support/integration-guides/compatible-vpn-routers/)’s website under Product documentation.

You will find a list of compatible VPN firewalls/routers and the corresponding configuration guides on our website at:

<https://www.thegreenbow.com/en/support/integration-guides/compatible-vpn-routers/>.

You can download a demo configuration and open a test tunnel by following the instructions on our website at:

<https://www.thegreenbow.com/en/frequently-asked-questions/#deeplink-4091>.

You will find more information about TheGreenBow products on our website:

<https://thegreenbow.com/>.

## 19 OpenSSL license

OpenSSL is licensed under the Apache License 2.0 reproduced below.

Apache License  
Version 2.0, January 2004  
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.



3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS



## **20      Contact**

### **20.1      Information**

All the information on TheGreenBow products is available on our website:  
<https://thegreenbow.com/>.

### **20.2      Sales**

Phone: +33.1.43.12.39.30

E-mail: [sales@thegreenbow.com](mailto:sales@thegreenbow.com)

### **20.3      Support**

There are several pages related to the software's technical support on our website:

#### **Online help**

<https://www.thegreenbow.com/en/support/online-support/>

#### **FAQ**

<https://www.thegreenbow.com/en/frequently-asked-questions/>

#### **Contact form**

Technical support can be reached using the form on our website at the following address: <https://www.thegreenbow.com/en/support/online-support/technical-support/>.

**Secure your connections**  
in all circumstances